**四川大学**

**工程硕士专业学位论文**

**题 目 基于深度学习的端到端异常入侵检测方法研究**

**作 者 \*\*\* 完成日期 2022 年 3 月 30 日**

**培养单位 四川大学**

**指导教师 \*\***

**指导教师**

**工程领域 计算机技术**

**授予学位日期 2022 年 月 日**

基于深度学习的端到端异常入侵检测方法研究

计算机技术领域

研究生\*\*\* 指导老师 \*\*

异常入侵检测是网络安全系统的重要组成部分，它对进入网络的流量进行检测和监督，一旦发现异常活动，系统就迅速做出响应。异常入侵检测通过对正常行为建模来区分识别异常的网络行为和正常的网络行为，具有识别新型攻击的能力，对于在当前愈加复杂多变的网络环境下构筑网络安全具有重要意义。

数据是影响算法性能的重要因素，异常入侵检测使用的数据是从网络上捕获的网络流量数据。根据对网络流量数据的处理方法，可以将异常入侵检测方法分为两种。一种是利用专家知识，统计学来提取人工特征，然后执行检测的检测方法：将通信会话持续时间、数据包个数、流量大小、每秒字节数、具有相同源地址的会话个数等统计量作为表示数据集中单个样本的特征，使用传统机器学习算法如SVM,决策树，贝叶斯等算法构建检测模型；另一种是直接使用网络流量数据构建检测模型的方法：随着深度学习的发展，以及硬件进步带来的计算资源的丰富，直接使用原始网络流量数据作为模型输入构建端到端的模型算法成为可能，且该方法无需人工提取特征，是当前入侵检测领域的研究热点。

使用原始网络流量数据构建端到端的入侵检测模型存在三个问题。一是网络流量数据无法直接使用，从网络上捕获的pcap文件，无法直接使用，还需要按照数据连接的定义将其分割，并以某种形式存储，形成可用于深度学习的数据集。针对此问题，需要研究者处理分割pcap文件，合理采集样本，构建可用的深度学习数据集；二是当前基于深度学习的入侵检测算法准确率较低；三是模型可解释性差，当前的入侵检测模型通常没有做可解释性方面的工作，对于提出的模型的判断机制是否符合人类理解没有给出结论，人类对于模型为何做出某种决策无法理解，导致模型完全是一个黑箱模型，导致人类缺乏对模型的信任。对于上述三个问题，本文工作如下：

一、构建了基于CICIDS-2018的会话样本数据集CICIDS-sessions。本文首先具体分析CICIDS-2018原始网络流量数据pcap文件，在分析每个样本的数据包大小，通信持续时间等统计信息的基础上，提出了一种流量分割方法，对pcap格式的数据文件进行流量分割，通过提取两个主机之间通信的多个数据包组成单个会话作为一个数据样本，以矩阵的形式进行存储，构建了一个可直接使用的全面的包含80万个样本15种类别的会话样本数据集，该数据集也是本文接下来提出的检测算法所使用的数据集。

二、提出一种基于深度学习的端到端的异常入侵检测算法。该算法模型包括四个不同模块，核心思想是利用CNN提取空间特征，利用循环神经网络提取序列特征，利用两者的融合特征进行入侵检测。以CICIDS-sessions提供的深度学习会话样本作为输入，模型流程如下：通过将会话样本的数据包融合形成二维矩阵送入卷积模块,将单个数据包作为一维序列送入循环模块,分别提取特征，利用注意力机制为卷积模块输出的不同区域分配不同权重，最后将注意力图和循环模块的融合特征送入分类层执行入侵检测。该算法无需人工提取特征，具有实时检测的优点。实验证明，该算法优于使用人工特征集方法。

三、基于全局积分梯度的模型可解释性方法。模型得到预测后通过对输入的某个特征求导，得到的梯度可以反映输入某个维度特征对预测的贡献度。本文提出了计算全局积分梯度的归因方法，该方法可以计算模型对于样本某个类别的全局归因分布（模型输入对于模型输出的重要性分数），而不是单个样本的局部归因分布，以此更好的解释模型。通过该归因分布，可以得到二维会话图像的可视化归因图，理解模型如何利用会话图像；得到数据包的重要性排名，理解模型每次分类中重点关注的数据包，最后尝试将重要性分数对应字节和数据包网络协议的格式对应，论证是否有强相关性。

关键词：异常入侵检测 深度学习 流量分割 模型可解释 全局积分梯度

**An end-to-end original traffic intrusion detection method based on deep learning**

Presented for MEng Degree

**Subject: Computer Technology**

**Postgraduate: \*\*\* Supervisor: \*\***

**Abstract**

**Keywords** Anomaly Intrusion Detection, deep learning, Traffic segmentation, model interpretable, Global integral gradient

**目 录**

[第1章 绪论 1](#_Toc102139290)

[1.1 研究背景及意义 1](#_Toc102139291)

[1.2 国内外研究现状 2](#_Toc102139292)

[1.2.1 传统机器学习算法阶段 3](#_Toc102139293)

[1.2.2 深度学习算法阶段 4](#_Toc102139294)

[1.2.3 本文工作与现有研究 5](#_Toc102139295)

[1.3论文研究内容与组织结构 6](#_Toc102139296)

[第2章 入侵检测相关技术研究 8](#_Toc102139297)

[2.1 传统机器学习 8](#_Toc102139298)

[2.1.1 决策树 8](#_Toc102139299)

[2.1.2 支持向量机 11](#_Toc102139300)

[2.1.3 朴素贝叶斯 15](#_Toc102139301)

[2.1.4 随机森林 16](#_Toc102139302)

[2.1.5 AdaBoost 18](#_Toc102139303)

[2.2 深度学习 20](#_Toc102139304)

[2.2.1 卷积神经网络CNN 20](#_Toc102139305)

[2.2.2 循环神经网络RNN 24](#_Toc102139306)

[2.3 本章小结 27](#_Toc102139307)

[第3章 基于CICIDS-2018的会话样本数据集构建 28](#_Toc102139308)

[3.1 数据集构建方法 28](#_Toc102139309)

[3.1.1 CICIDS-2018网络流量数据最大数据包字节数，最大数据包个数 30](#_Toc102139310)

[3.1.2 最大数据包字节数，最大数据包个数 30](#_Toc102139311)

[3.1.3 流量分割方法流程 32](#_Toc102139312)

[3.2 会话样本数据集构建实验 34](#_Toc102139313)

[3.2.1 准备工作 34](#_Toc102139314)

[3.2.2 分割结果 36](#_Toc102139315)

[3.2.3 样本标签 36](#_Toc102139316)

[3.3 会话样本数据集分析 37](#_Toc102139317)

[3.3.1 可视化分析 37](#_Toc102139318)

[3.3.2 数据包数目，数据包字节，会话持续时间 41](#_Toc102139319)

[3.4 本章小结 44](#_Toc102139320)

[第4章 基于深度学习的端到端的异常入侵检测算法 45](#_Toc102139321)

[4.1 算法 45](#_Toc102139322)

[4.1.1 模型整体结构 45](#_Toc102139323)

[4.1.2 卷积模块 46](#_Toc102139324)

[4.1.3 循环模块 47](#_Toc102139325)

[4.1.4 注意力模块 48](#_Toc102139326)

[4.1.5 分类模块和总结 50](#_Toc102139327)

[4.1.6 损失函数和优化算法 51](#_Toc102139328)

[4.2 实验设计 54](#_Toc102139329)

[4.2.1 实验目的和步骤 54](#_Toc102139330)

[4.2.1 实验环境和学习率设置 55](#_Toc102139331)

[4.2.3 数据和参数设置 56](#_Toc102139332)

[4.3 实验结果与分析 58](#_Toc102139333)

[4.3.1 评价指标 58](#_Toc102139334)

[4.3.2 基于会话样本的入侵检测实验 60](#_Toc102139335)

[4.3.3 基于人工特征的入侵检测实验 66](#_Toc102139336)

[4.4 本章小结 69](#_Toc102139337)

[第5章 基于全局积分梯度的模型可解释性方法 70](#_Toc102139338)

[5.1 IG积分梯度 70](#_Toc102139339)

[5.1.1 归因定义 70](#_Toc102139340)

[5.1.2 两个基本公理 71](#_Toc102139341)

[5.1.3 IG计算方法 73](#_Toc102139342)

[5.2 可解释方法实验 74](#_Toc102139343)

[5.2.1 全局积分梯度 75](#_Toc102139344)

[5.2.2 数据包重要性排名 76](#_Toc102139345)

[5.2.3 数据包格式关键位 77](#_Toc102139346)

[5.3 可解释方法实验结果分析 78](#_Toc102139347)

[5.3.1 结果可视化图 78](#_Toc102139348)

[5.3.2 结果理解 86](#_Toc102139349)

[5.4 本章小结 87](#_Toc102139350)

[第6章 总结与展望 88](#_Toc102139351)

[6.1 总结 88](#_Toc102139352)

[6.2 展望 89](#_Toc102139353)

[参考文献 90](#_Toc102139354)

[附录1 XXXXXX 94](#_Toc102139355)

[附录2 XXXXXX 94](#_Toc102139356)

[作者在读期间科研成果简介 95](#_Toc102139357)

[致 谢 96](#_Toc102139358)

[声 明 97](#_Toc102139359)

[学位论文版权使用授权书 98](#_Toc102139360)

1. 绪论

本章将阐述论文选题的背景，和异常入侵检测的研究意义，并对本文的研究内容和组织架构进行概述。

1.1 研究背景及意义

随着互联网的快速发展，互联网上的活动涉及到了社会生活的方方面面，互联网已经成为了人类社会的基础设施，但是伴随着互联网上经济财富的快速增长，个人，企业，政府都面临着越来越多来自互联网上的恶意组织和个人的网络攻击。而且随着互联网技术的发展，要应对的攻击问题更加多样，复杂而且变化速度极快，建立健全完善的网络安全机制已经刻不容缓。异常入侵检测技术是网络安全防线的先锋，它使用基于机器学习的技术检测与当前系统通信的网络流量是否是正常流量，是当前网络安全领域的研究热点。

据2020年我国互联网网络安全态势综述报告显示[1]，我国全年一年捕获恶意样本数量超过4200万个，涉及多达34.8万个恶意软件家族；移动互联网恶意程序数量逐年上升，至2020年已经高达303万个；智能联网设备恶意程序数量捕获样本341万个，同比增长8.5%；国家信息安全漏洞共享平台2020年收录漏洞20704个，收录安全漏洞数量呈上升趋势；分布式拒绝服务攻击日渐呈现大流量攻击趋势，攻击资源数量逐年增加；云平台上的网络安全也出现了许多新问题：利用云平台资源发起的DDoS攻击占国内DDoS攻击事件的81.3%，云平台承载的恶意程序数量种类占国内全网的83.3%，依靠云平台资源的便利性，木马，僵尸网络等恶意代码的传播速度远超过去。

面对如此多样的通过互联网进行的攻击，目前针对基于网络的入侵检测有两种防御思路。一种是通过建立一系列的规则来定义每一种恶意网络活动，然后通过规则匹配的方式来检测网络入侵。当前比较受欢迎的基于规则的开源入侵防御系统是Snort3[2]，它可以支持用户自定义规则来检测某种恶意网络活动。这种方式是一种基于黑名单的思路，它只能检测已有的攻击类型，对于新型攻击类型（未发现的漏洞）无能为力[3]。另一种方式就是本文所研究的入侵检测方式，即异常检测[4]，它通过机器学习的方法构建一个模型，从网络行为和系统日志中提供的特征来学习识别正常数据和异常数据，具备检测潜在攻击的能力。该能力的来源是对正常网络行为和异常行为进行了建模，分析了正常网络行为和异常网络行为的差异，因此在面对新的网络攻击时，根据和正常网络行为的差异程度就能以概率的结果猜测是正常或是异常的网络行为。

因此，异常入侵检测算法定义为检测算法对用户正常行为进行建模，可以检测异常的网络流量。采用这种思路进行入侵检测有如下优点：一是可以检测新型攻击（训练数据集没有包含的攻击），通过判别流量是否是正常用户行为模式下的流量来识别异常流量，即使出现的是新的攻击类型，只要该流量特征不符合正常用户行为模式，就会被识别为异常；二，可以应对零日攻击（专指在当前没有被公布的漏洞），增加系统抵御未知漏洞或攻击的能力。然而，异常入侵检测也有如下难点：一是用户正常行为建模困难，需要完善的用户行为数据，数据收集和处理是重点工作。我们可以想象该工作的困难程度，现如今网络上的活动愈加广泛，用户的网络行为也逐渐变得复杂；二是对于已有的攻击类型检测的准确率在整体上没有基于规则的检测算法准确率高，这是很容易理解的，对于已有的攻击类型，只要进行了全面分析，提取攻击特征码，那么这类攻击基本难逃检测，这也是为什么网络安全应用程序需要及时更新病毒数据库的原因；三是误报率较高，误报率较高的原因也是明确的，即对正常网络用户行为的建模不能达到100%，没有办法像针对某一类攻击那样提取其特征码，实现近乎百分百的检测率。

采用异常入侵检测能够检测新型网络攻击具有非常吸引人的优点，但也有不少困难需要面对,如怎么样获取全面而足够代表性的数据，怎样选取和构建性能良好的算法，是否采用人工特征还是使用表示学习得到的自动特征。现如今深度学习技术快速发展，将深度学习应用于异常入侵检测成为流行趋势，并且研究表明使用深度学习技术可以显著降低异常入侵检测技术对于用户行为建模困难，准确率低，和误报率高的缺点。在如今网络安全形式日趋严峻，网络攻击行为多样且变化速度快的背景下，可以应对新的攻击的异常入侵检测就显得格外重要。

1.2 国内外研究现状

对于异常入侵检测，其整体流程如图1.1。

异常入侵技术使用的数据主要包括网络流量数据和系统生成的日志。



图1.1传统机器学习与深度学习的异常入侵检测算法流程对比

Fig. 1.1 The Comparison of abnormal intrusion detection algorithm process between traditional machine learning and deep learning

异常入侵检测技术的主要经历了两个阶段，第一阶段是基于传统机器学习技术，如图1.1上半所示流程，从网络流量数据和日志中经过特征过程生成特征集，需要人工设计和提取特征并且这阶段任务得到的特征是否合适完全取决于专家知识，特征的效果是不确定的，需要在模型进行测试得到相关性能评价指标后才能确定。后续可能需要不断调整特征集，不能在训练过程中实时调整特征集，只能在经过完整的训练得到一个模型后评判特征集的好坏，因此该项工作是和模型检测任务是割裂的，不是一个统一的整体，不是端到端的架构。

另一阶段是使用深度学习算法，其流程如图1.1下半所示，算法架构于数据之上，一端是数据，一端是模型和入侵检测任务，模型特征提取和模型检测任务是高度相关的，两者是一个统一的整体，是不可分割的，采用端到端方式完成入侵检测。深度学习有一个重要优点是表示学习的能力，它可以通过损失函数优化，后向传播梯度数据，在不断迭代中学习到对模型检测任务最为合适的特征，在训练过程中不断改变学习到的特征，这种通过表示学习的方法得到的特征可以称之为自动特征。

基于深度学习的自动提取特征有着明显的优点，即无需人工构建特征集，模型特征提取和模型入侵检测任务是一体的，两者在训练过程中不断迭代进步，当模型入侵检测任务性能指标表现良好时，证明提取的特征也是优秀的。

接下来本文将详细介绍这两阶段国内外的研究现状，并分析现有研究和本文工作的联系。

1.2.1 传统机器学习算法阶段

基于网络的异常入侵检测的原始数据对象是网络流量数据和日志。网络流量通常是通过网络流量捕获软件收集记录而成，一般是pcap类型的文件。对于整体的文件根据某种定义进行分割，比如一个TCP连接是一个记录，通过专家知识和统计学进行特征工程得到基于网络流量数据的特征数据集。入侵检测中使用最广泛的基准数据集是KDDCUP-99数据集[5]，它是在模拟美国空军的局域网上收集7天数据所后构建的，总共4898431条记录，每条记录41个特征，多达23个攻击类别，最初提出是用于第三届互联网知识发现和数据挖掘竞赛。对于此数据集我们可以轻松获取，另外通过当前最受欢迎scikit-learn开源机器学习库也可以获取[6]。更多基于网络的入侵检测特征数据集可以参考文献[7]。

基于特征数据集，研究者采用了许多机器学习算法构建模型进行入侵检测。在引文[8]中作者分别使用支持向量机（support vector machines，SVM）和神经网络（Neural Network）的方式构建入侵检测模型，论文作者所采用的神经网络模型是多层前馈网络（或者说是多层感知机），并不是之后发展出的更加复杂和专业化的神经网络，如卷积神经网络CNN,循环神经网络RNN。研究表明SVM模型算法准确度更高（均为二分类，SVM为99.50%，Neural Network最高为99.25%）同时在训练模型所耗费的时间上，SVM速度更快（17.77s比18min）。

K最近邻算法(k-Nearest Neighbor，KNN) 是一种高效简单的分类算法，引文[9]详细研究了KNN算法和自己改进的基于KNN的CANN算法在异常检测上的性能表现，在经过特征选取的9维特征集和19维特征集上，基于KNN的CANN算法测试集准确率为99.76%与99.46%，误报率为0.003%与2.95%。在引文[10]中，作者使用单层决策树Decision Tree作为AdaBoost算法的弱分类器，进行几轮迭代学习，最后组合弱分类器成为最终用于检测任务的强分类器，在KDDCUP-99测试集上准确率是90.47%，误报率是0.665%。需要指出的是，现如今算法快速发展，使用scikit-learn机器学习库的决策树算法，可以在KDDCUP-99数据集上达到99.99%的准确率和0.01%的误报率，但这也是建立在以前研究者在算法改进和参数探索研究的努力，和当前计算机硬件性能提升可以支撑更加复杂和大量的计算基础之上。

在传统机器学习中随机森林（RandomForest）作为一种集成学习也应用广泛。在引文[11]作者详细研究了使用各种组合的随机森林算法参数（决策树的个数，特征选择方法）在KDDCUP-99子集上执行入侵检测任务的结果，在选择子决策树个数15，特征数据集特征维数为34的参数情况下，随机森林算法达到了最佳效果：准确率94.2%,误报率1.1%。值得一提的是，使用如今的scikit-learn库的默认RandomForestClassifier分类器在KDDUP-99改进数据集NSLKDD上可以达到99.99%准确率和0.01%的误报率。

在异常入侵检测领域上，研究者们几乎尝试了所有的传统机器学习方法，在较为古老的基准数据集KDDCUP-99达到了99.99%的准确率。这些实验表明对于在过时网络上的老旧攻击类型，基于专家知识建立特征集，使用当前发展较为完善的传统机器学习方法进行入侵检测是非常棒的方法。但是当前互联网的网络早已不同于过去，尤其是在当前云计算应用越来愈广泛的时代，流量巨大，攻击类型日益多变，传统机器学习和人工特征集的方法已经不能适应时代需求了，正如现如今在计算机视觉领域，深度学习已经几乎全面取代了传统机器学习方法。

1.2.2 深度学习算法阶段

2012年，深度卷积神经网络AlexNet[12]以低于第二名10%以上错误率的巨大优势获得ImageNet的冠军，深度学习在学术界吸引了越来越多的人展开研究，将深度学习技术应用于异常入侵检测也成为这个领域研究的热点。

对于应用深度学习技术，也有两种不同的方法。其一就是基于之前特征数据集的深度学习应用，即算法的输入是每一条记录的特征值（通常经过处理后表现为一维的特征向量），输出是模型预测的类别。另外一种就是直接将深度学习算法架构于网络流量数据上（使用tcpdump或类似工具捕获的真实流量数据），一端就是网络流量数据，一端是样本攻击类别输出，是一种端到端的算法。本文研究的重点就是端到端的深度学习算法。下面介绍关于这两种不同思路的研究情况。

基于特征数据集：在引文[13]中，作者研究了只采取通常RNN循环单元（不包括LSTM，GRU）通过调整隐藏层节点数量和学习率对模型性能的影响，实验在KDDCUP数据集上进行，实验表明使用RNN在该数据集上获得了不亚于传统机器学习算法的良好效果。在引文[14]中作者对比了使用LSTM和GRU模型的性能表现，实验表明相比LSTM网络结构，GRU更加适合入侵检测任务。在引文[15]中作者使用CNN+RNN，其中CNN是一维卷积网络，RNN包括LSTM和GRU，在数据集KDDCup99上做了实验，实验证明CNN能够提取特征数据的更高层抽象特征来提高模型性能。相比在使用基于规则的数据集上的机器学习算法，深度学习算法表现的性能没有在计算机视觉领域表现的出色，但也十分优秀。

基于原始网络流量数据：引文[16]中，Wang等人还将网络流量数据转化为单一通道黑白图片，使用卷积神经网络做入侵检测，在自建数据集（组合了一部分公开数据集和自己生成的数据）上达到平均准确率99.41%良好效果，在此次工作中，作者还对如何表示网络流量做了实验，实验证明使用所有会话包的双向流（即会话）会取得最佳实验效果。在引文[17]中提出使用一维卷积网络将网络流量视为一维输入做加密流量识别取得了较好的成绩。也有研究者将网络流量的数据包作为检测对象，使用滑动窗口的形式将一个数据包的字节分为多个字节序列送入RNN中，训练模型用于入侵检测，实验表明该方式取得了不错的效果。

另外，如今图神经网络兴起，在社交网络，分子合成，用户推荐预测等领域表现出优秀的性能，也有研究者开始探索图神经网络在入侵检测中的应用，在引文[18]中作者将主机和会话表示为节点，边表示为这主机之间的通信关系，使用节点分类的图神经网络模型来进行入侵检测，需要指出的是该论文也是架构于特征数据集之上。另外也有研究者[19]通过某个函数来衡量某两条连接之间的相似关系来构建边，如果两天边执行某种运算得到某个分数大于某个阈值则视为两条连接之间有边，从而应用图神经网络进行异常入侵检测。将主机视为节点，关注网络的拓扑关系，应用图神经网络也是一种方向[20]。当然在其他如物联网IoT领域，由于存在各个物理节点之间天然的物理关系，应用图神经网络会更加自然且应当使用这种技术。

综合上述深度学习算法在异常入侵检测上的研究，可以发现该技术阶段的主要创新集中在两个方面：一是使用深度学习算法代替传统机器学习算法，在人工特征集上训练模型。该方法在引文13，14，15上都得到了实验证明，该方法是可行的，并且性能优秀。二是充分利用深度学习技术在表示学习上自动提取特征的能力，通过将原始网络流量数据处理成合适的样本形式（通常是某种形式的张量）使用深度学习技术神经网络建立端到端的检测模型，即模型特征提取和模型决策任务是一体的架构。方面二的创新具有明显的优点和缺点，缺点是将原始流量数据出何种格式的张量（这有点类似与计算机视觉中图片的分辨率）需要经过充分的统计分析和实验，优点就是本文之前一直强调的无需人工设计特征，模型特征提取和模型决策任务为一个整体，为端到端架构。

对于不同的入侵检测算法，如何评估算法的好坏，以及什么样的数据集是优秀的，也是一个该领域研究中的关注重点。研究者S. García等[21]通过在一个大型的真实的有标记的僵尸网络数据集对比了三种不同的僵尸网络入侵检测方法，建立了更加合适的评价指标，实验表明好的比较方法有助于改进算法，构建更好的数据集，因此在入侵检测领域，在同一个数据集上进行不同方法的详细比较可以有效的评估算法的性能，这有助于入侵检测领域的进步，也是本文进行对比实验的指导，即充分在各个评价指标上衡量模型的性能。

1.2.3 本文工作与现有研究

从上述现有研究工作分析可知，基于传统机器学习算法的方法已经不能满足当前入侵检测的需要。面对日益复杂的网络安全环境，种类和数量都日益增多的攻击类型，传统的通过人工专家提取特征，使用传统机器学习算法进行入侵检测方法，不能应对越来越多的新型攻击（尤其是零日攻击）,需要重新设计人工特征。而随着深度学习技术的兴起，研究者开始运用深度学习的技术，自动提取特征，并且在该领域取得了不错成果。

深度学习的技术特点是能够从原始复杂的数据中自动学习特征。从原始网络流量数据中自动学习特征，然后使用神经网络进行预测，是本文的核心思路，因此本文研究充分吸收了入侵检测领域在深度学习方面的成果，提出了一种端到端的入侵检测算法，为本文第4章的成果。要实际运用此核心思路，需要解决一个重要问题，就是如何处理这些原始的网络流量数据，这是本文第3章讲述的内容，一种基于CICIDS-2018的会话样本数据集。同时使用深度学习也面临着一个重要问题，就是可解释性的问题。基于专家知识的人工特征固然费时费力，而且面对新型攻击往往需要重新设计特征，但是每个特征都是人类可理解的，如流持续时间，登录次数等，但是深度学习得到的特征人类往往难以理解，而这是本文第5章解决的问题，提出了基于全局积分梯度的可解释性方法，用于理解模型，解释模型的决策机制。

1.3论文研究内容与组织结构

本文研究内容为基于深度学习构建的端到端的异常入侵检测技术，主要研究将算法直接构建于网络流量数据之上，挣脱人工特征数据集的束缚，通过深度学习技术进行端到端的表示学习，从网络流量数据学习出对高级抽象特征用于入侵检测分类。本文研究内容和创新点如下：

1：基于CICIDS-2018的会话样本数据集，提出一种流量分割算法，将网络流量数据处理成一定格式的张量样本，构建了一个可以用于深度学习的，已标记好的会话样本数据集。

2：提出了一种基于深度学习的端到端的入侵检测算法：该算法架构于网络流量数据之上，充分考虑了会话样本中各个数据包的序列关系，整体会话样本的结构特征，并且在1所述的数据集上做了全面实验，实验证明该算法拥有优秀的性能，使用网络流量数据构建端到端深度学习模型是完全可行的，不需要人工参与手动提取特征。

3：基于全局积分梯度的模型可解释性方法。模型可解释性：在每次对一个会话进行识别中，是会话中哪个数据包占据重要作用呢？以及经过深度学习提起的特征中是哪几个特征在分类中具有决定作用？本文提出了计算全局积分梯度的归因方法，该方法可以计算模型对于样本某个类别的全局归因分布，而不是单个样本的局部归因分布。通过该归因分布，可以得到二维会话图像的可视化归因图，理解模型如何利用会话图像；得到数据包的重要性排名，理解模型每次分类中重点关注的数据包，最后尝试将重要性分数对应字节和数据包网络协议的格式对应，论证是否有强相关性。

本文组织结构如下：

第1章 绪论。阐述了论文的研究背景和意义、国内外研究现状以及论文的主要研究内容和整体组织结构。

第2章 相关技术研究。阐述异常入侵检测用到的相关技术，主要包括两部分，其一是简要介绍传统机器学习算法，然后重点介绍本文研究使用的深度学习算法。

第3章 基于CICIDS-2018的会话样本数据集构建。基于ids-2018的网络流量数据集构建。CICIDS-2018只提供网络流量捕获文件，如果要用于训练需要定义如何进行流量分割，本文通过研究数据集数据包字节分布规律，时间要素等，提出一种会话流量分割方式，并将该网络流量数据文件构建成一个可直接用于深度学习，可用于端到端训练的，存储成张量形式的样本数据集。

第4章 基于深度学习的端到端异常入侵检测算法。提出一种直接使用网络流量数据上的一种端到端的深度学习算法，该算法学习会话样本的整体特征和会话样本中各个数据包的序列特征用于异常入侵检测。在与基于人工设计特征的传统机器学习方法的对比实验中，验证了算法的优秀的性能。

第5章 基于全局积分梯度的模型可解释性方法。模型可解释性的目的就是找到输入相对于输出的归因分布（更具归因方法，给予输入所有特征元素一个重要性分数）。本文提出一种计算全局积分梯度的归因方法，该方法将计算单个样本的归因扩展到全局样本的归因，根据全局积分梯度，本文对第4章模型进行解释，得到了会话图像输入的可视化归因图，数据包的重要性排名，论证了将重要性分数对应字节和数据包网络协议的格式对应的思路是否可行。

第6章 总结与展望。总结论文工作，针对论文工作不足，提出设想，展望未来工作。

1. 入侵检测相关技术研究

异常入侵检测通常表现为如下机器学习问题，给定训练样本**x**={x1, x1,…, xn}，其类别y={正常，异常},从训练样本中学习一个分类器，输入样本特征向量**x**，预测其是正常或异常样本。训练出的分类器由于对正常行为模式进行了建模，可以根据样本和正常行为模式之间的差异来识别异常样本。

本章主要介绍异常入侵检测技术相关知识，异常入侵检测的定义，以及研究者使用传统机器学习技术和深度学习技术进行异常入侵检测研究的方法，并对每个方法的优点和缺点进行分析。最后，本章的重点是本文所使用的深度学习中的卷积神经网络与循环神经网络。

在本节中，我们约定如**w**这样的加粗字母代表矩阵或者向量。

2.1 传统机器学习

机器学习是涉及概率论，统计学，微积分，线性代数，脑科学等多领域学科的一门科学，机器学习可以让计算机从数据中学习知识，从而做出决策判断，让机器像人一样思考，是对人类智能的一种模拟。算法在学习中会通过改变自身参数，优化结构来不断提高性能表现，类似于人类的学习行为，因此得名机器学习。深度学习是机器学习的一个分支，本节主要介绍传统机器学习算法。

传统机器学习算法分为有监督算法和无监督算法，但是无论如何分类，机器学习终究是数据驱动的，数据的质量对于模型的影响非常关键。而在入侵检测领域主要使用的是监督式学习算法，关于无监督算法，Tommaso Zoppi详细介绍了采用元学习[22]的方式，通过两层堆叠（下层是无监督学习算法部分，上层是监督学习算法部分）的结构，提出了一种结合无监督学习和监督学习的两种模式的算法，有效结合两种算法的优点，加强了模型检测未知攻击的能力[23]，无监督学习通常用于表示学习，一般和监督学习结合使用。经典的机器学习分类算法包括决策树[24]( CART[25])，支持向量机[26]，贝叶斯[27]，经典的集成学习包括随机森林[28]，AdaBoost[30]等。本节主要介绍上述算法。

2.1.1 决策树

决策树[24]是一种监督学习算法，它从数据特征里学习简单的决策规则，每一个非叶节点都表示一个特征向量中的一个维度xi，根据样本在该特征维度的某种值来划分树枝走向，从根节点开始一步一步划分，最终将所有样本归为某个类别，最后就是一个树形的决策图。算法在每一次迭代中根据某种评价指标选择某个特征作为节点来划分样本集。

有个简单的例子可以说明决策树的思考过程。小明最近一直在烦忧自己未来的职业选择，最后他决定根据这一学期的考试成绩，兴趣爱好，体育运动这三项来决定自己是当科学家还是运动员。具体决策过程如图2.1所示：



图2.1 小明的职业决策过程

Fig. 2.1 The career decision process of XiaoMing

在图2.1中，圆形节点表示小明决策未来发展方向向量的一个特征维度，圆角矩形代表叶子节点表示最终的样本类别。

决策树算法有许多优点：1，易于理解和解释，并且可如图2.1进行可视化；2，所需数据样本少，模型做出预测时花费（决策判断次数）是训练样本的log数次；3，模型是白盒模型，非常容易用统计学测试模型可靠性[24]。

决策树缺点如下：1，容易生成复杂的树，即过拟合，需要剪枝，限制树最大深度；2，当前算法模型采用启发式算法得到局部最优解，不能得到全局最优决策树；3，对于某些概念如XOR,奇偶校验难于学习，且对于类别不平衡比较敏感。

当下，比较常用的决策树算法是ID3,C4.5,CART,这几种算法基于不同的启发式算法选取某个特征作为切分节点，将样本划分到不同的分支中，在各个分支中不断迭代选取特征节点，直到决策树生长到最大高度，最后可以使用剪枝提高泛化能力。

1. ID3和C4.5

ID3使用信息增益选择最大信息增益的特征作为分裂依据。要了解最大信息增益，首先需要学习信息熵。对于样本集合D，样本类别数K,其信息熵如公式2.1：

  （2.1）

其中：

表示样本中属于第k类的子样本集。

表示该子样本集合内元素个数。

表示总样本集合内元素个数。

对于特征A，可以根据特征A将样本划分为n个子集，则特征A对于集合D的条件信息熵为式2.2：

 （2.2）

其中：

Di表示D中特征A取第i个值时的样本子集。

Dik表示Di中样本是第k类的样本子集。

在求得上述两个值后，就可以求得所有的关于某个特征A的信息增益，如式2.3：

 （2.3）

ID3算法选取信息增益最大的A特征来最为第一个切分节点，在之后的样本中删去A特征，然后迭代计算信息增益，不断在特征子集中选取切分特征，最后直到决策树生长到最大高度，特征选取完毕。

但是ID3有一个缺点，就是ID3倾向于取值数多的特征.列如对于特征A1和A2, A1取3个值，A2可以取5个值，那么计算得到的信息增益A2>A1。这样看似比较合理，但是实际应用中却让算法表现很差。比如，在判断一个事情是否有趣时，对于特征参与人数来说，其取值会非常多，但是用这个特征作为第一个切分节点来划分样本，生成决策树后，该决策树泛化能力很弱，无法应对训练集以外的情况。

为此，我们需要对样本取值较多的特征进行惩罚，由此产生了另一种算法C4.5。其采用信息增益比作为选择切分特征的准则，如式2.4：

 （2.4）

其中，就是D关于A的取值熵，如式2.5，也就是惩罚系数，可以清晰看到，取值越多的特征A，该值就会较大，从而得到较小的信息增益比。

 （2.5）

此外，C4.5还改进了ID3不能处理连续变量的缺点，方法是将连续变量划分为几个不同的值域区间，从而将连续变量转换为具有多个类别的离散变量。

1. CART 

CART（Classification and Regression Trees）使用基尼系数作为构建准则选择切分特征[25]。公式2.6如下：

 （2.6）

在特征A将D划分的情况下，特征A基尼系数为式2.7：

 （2.7）

CART算法每次选取基尼系数最小的特征A和对应的切分点作为划分节点。值得注意的是CART算法每次切分只会将样本集切为两个类别，即CART决策树是二叉树。列如对于特征A，有三个类别A1,A2,A3,分别计算采用切分点A1,A2,A3的三个基尼系数，假设最小基尼系数所在切分点为A1,则将样本划分为{A1},{A2,A3}两个子集，并且在接下来的迭代中，该特征仍然可以作为切分节点，如下次可以划分为{A2},{A3}。

本文使用算法为CART。表2.1总结了三种决策树的不同。

表2.1三种决策树算法对比表

Table2.1 Three Decision Trees Comparison table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **算法** | **支持变量类型** | **树结构** | **计算类型** | **任务** |
| ID3 | 离散型变量 | 多分支，特征不复用 | 对数运算 | 分类 |
| C4.5 | 离散型和连续型变量 | 多分支，特征不复用 | 对数运算 | 分类与回归 |
| CART | 离散型和连续型变量 | 二叉树，特征复用 | 非对数运算 | 分类与回归 |

2.1.2 支持向量机

支持向量机（Support vetor Machine，SVM）[26]是一种监督类型的分类机器学习算法，它通过寻找作为决策边界的最大间隔超平面将样本进行分类。对于决策边界可以分为硬边界和加入了松弛变量的软边界。对于线性可分的问题可以使用线性SVM，而对于线性不可分的问题，可以通过核函数将样本映射到高维空间，寻找高维的超平面将样本分到平面上下。对于线性可分问题的svm算法，可以看图2.2所示：



图2.2 SVM关于线性可分离散样本示意图

Fig. 2.2 Schematic diagram of SVM about linearly separable scattered samples

 如图所示，黑色实心点为负样本，白色空心点为正样本。样本集为:{**xi,**y},i=1,2,3,…,n，样本类别y{+1,-1}，**xi**Rd。SVM寻找这样一个决策边界-超平面H0：，所有样本点到超平面的距离最远，如图中所示间隔最大的超平面（实心直线）,而在实际运算中，我们要寻找如图中H1,H2两个超平面,公式如下所示：

Hyperplane1，式2.8:

 （2.8）

Hyperplane2，式2.9:

 （2.9）

其中**w**是超平面的法向量，b是截距，不等式右边取1是为了便于运算，取其他任何值都不影响优化问题的解。而落在这两个超平面上的样本是支持向量。

合并上述两个公式得到公式2.10：

 （2.10）

最大化距离d=margin(H1和H2平面之剑的距离)，如式2.11：

 （2.11）

其中可以由点到H0超平面的距离公式得到。

式2.11等价于最小化： 。

所以SVM的优化问题就转化为式2.12：

Minizing: ，subject to:  （2.12）

引入拉格朗日乘数,构造拉格朗日函数，优化问题就可以转化为式2.13：

Min: （2.13）

其中,为拉格朗日算子。 由此，svm的硬约束求解问题就转变为了最小化2.13式。

对于式2.13,在满足条件，和的条件下，Lp的最小值为一个定值,因此假设目标函数可以取得最小值：,且由于,因此LP<m,而此时只有参数未知，则为了LP接近最小值m，则需要求： ,所以式2.13表示的原始问题最后表达为式2.14：

 subject to: （2.14）

而满足KKT条件（取得最优值的解的条件，同时也有对偶问题的解的充要条件，也就是拉格朗日函数的在不等式约束下的泛化条件）的约束优化问题都具有强对偶性：求解最大最小参数的顺序互换不改变最优解，即式2.15：

 （2.15）

求导，式（2.13）表示的不等式约束优化问题的KKT（Karush-Kuhn-Tucker）条件为：

 （2.16）

 （2.17）

 （2.18）

 （2.19）

 （2.20）

将2.14和2.15代入2.13求得，最后原优化问题就变成对偶问题，如式2.21：

 （2.21）

即最后将问题转化为了对偶问题 [22]，求解对偶问题就是求解原问题，且更容易。对于软约束问题，我们引入松弛变量 ，使得部分样本点在margin之内，这样SVM优化就变为了式2.22：

 （2.22）

Subject to:  , 。

其中C为惩罚因子代表对错误样本的惩罚程度，由用户选取，C取值越大就代表不能容忍错误样本，则该优化变为硬约束问题，我们可以构造拉格朗日函数，在式子满足KKT条件下使用SMO(equential Minimal Optimization)求解。在软约束问题中，位于间隔内的样本点均是支持向量。

对于线性不可分的分类问题， 表示将样本点x映射到高维特征空间，由于在优化目标函数中样本点x通常要与另一个样本点进行内积，而如果采用先映射后内积的方式，运算量大，实际不可行，为此引入核函数K,来直接得到样本点映射到高维空间后的内积，即 ，则非线性SVM优化的对偶问题为式2.23：

 （2.23）

可以看到，核函数应用在原本样本点的内积上，可以直接使用核函数进行内积运算，而不需要先把样本点映射到新空间后再做运算，提高运算速度，以及节省内存空间。

常用的核函数有：线性核函数，多项式核函数，高斯核函数。

支持向量机的优点：1，最后学得的优化问题的解的参数只由支持向量决定，每次学习得到的预测函数稳定；2，严格的数学理论推导，可解释性强，白盒模型。缺点是：1，采用SMO算法，算法时间复杂度为O(n2);2,支持向量较多时，预测开销也越大，因此目前支持向量机对于大规模数据集表现不佳。

本文使用RBF(Radial Basis Function,g高斯核函数的变种)核函数的SVM。

2.1.3 朴素贝叶斯

朴素贝叶斯[27]假定样本**x**中各个特征两两之间相互独立，应用贝叶斯理论进行分类估计的一类算法，给定类变量y，特征向量**x**={x1,…, xn},

类变量有k种类别，为c1,…, ck，则有贝叶斯2.24：

 （2.24）

式中，由于各个特征xi之间相互独立，因此有，如式2.25：

 （2.25）

将2.25代入式2.24，最后朴素贝叶斯公式2.24变为式2.26：

 （2.26）

朴素贝叶斯就是求得一个概率分布函数P能够拟合数据集的经验分布，并取最大的概率所在的类别cm为预测类别，且由于是常数，最后求一个贝叶斯分类器就变成了如下问题，式2.27：

 （2.27）

在上述式子中，为训练集种类别y=cm的概率，可以由最大似然估计求得为训练集种类别y=cm出线的样本数除以总样本的个数，而对于后一个值,也称后验概率，需要用户指定，通常是根据样本特征的数据分布来选取也是算法需要学习的参数。

对于连续变量可以取高斯分布：

 （2.28）

式中，uy,也可由最大似然估计求出。

对于取多个值的离散分布，可以用引入平滑先验的多项式分布，概率函数如下：

 （2.29）

式中，Nyi为训练集中特征xi在类别为y=cm的样本中出现的次数， 为类别y=cm所有特征出现次数的总和。a=1为拉普拉斯Laplace平滑，a<1为Lidstone平滑。

特别的对于特征只取两个值的数据，二值特征向量，可以取后验概率为伯努利分布：

 （2.30）

朴素贝叶斯的优点：1，假设特征之间相互独立，特征之间相互解耦，可以将每一维特征描述为一维分布，有利于解决维度灾难。2，只需要少量数据就可以得到一个不错的分类器。缺点：1，实际中很多复杂问题往往不符合特征独立同分布的假设，因此朴素贝叶斯适用有限。

本文采用后验概率分布为高斯分布的朴素贝叶斯。

2.1.4 随机森林

随机森林（Random Forests）是由Breiman和Adele Cutler提出的一种基于bagging方法的集成学习算法[28]，该算法使用决策树作为基分类器，各个基分类器从有重复的不同样本集中独立学习生成，通过组合基分类器的平均预测概率得到最终预测结果（每个基分类器不单独，是一种Bagging(袋子)式集成学习。用随机森林得到分类器通常比单独的决策树方差要小，因此最后会得到一个较好的模型。

随机森林减小方差的秘诀在于随机。随机表现为两种方式：1，随机有放回的抽取样本，形成和原数据集数量一样的子数据集；2，在生成决策树，选择分裂特征时，不是从所有特征中选择最优准则的特征，而是随机选择m个特征，从这m个特征中选择最优特征。

根据这两个随机，随机森林中每颗树的生长过程如下[21]：

1.数据集样本数为N,则从原数据集中随机抽取N个样本，抽样方式为又放回抽样，因此形成的子数据集虽然样本个数为N,但是会包含重复样本，使用此子数据集来构造一颗决策树。

2.样本的特征维度为M，则选取一个远小于M的数m,从M个特征中随机选取m个特征，然后从这m个特征中选择最优特征作为分裂节点。M在随机森林生成过程中不变。

3.每棵树完全生长，不进行剪枝。

根据作者原始论文[24]，随机森林错误率主要由两个因素决定：1，任意两棵的相关性，相关性越强，错误率越高；2，每棵树作为一个好的分类器的强度，强度越高，随机森林错误率越低。

增加m会增加两颗树的关联性以及单颗树的强度，但是这样总体上的错误率就变得未知了，因此m有一个最佳范围，取这个范围内的m值，随机森林的错误率达到最低值。

具体的m值得求解可以通过袋外错误率（out of bag）进行验证。在上述构造子样本集的过程中，大约有1/3的样本没有被选取，这些落在袋子之外的样本就称之为袋外样本。在随机森林的算法中，根据上述树的建立过程建好k颗树后，对于个样本，考察没有作为子样本集中的分类结果，大约也是1/3的树没有使用该样本，按照多数投票的原则选择这1/3树的多数投票类作为算法对该样本的预测结果，遍历所有样本，计算被错误分类样本除以总样本数即为袋外错误率，计算几个不同的m值生成的随机森林的袋外错误率就可以确定最佳m值得范围。

由于这种袋外样本的存在，随机森林无需交叉验证或者使用额外的测试集就可以在随机森林构建过程中计算泛化误差（也就是上面介绍的袋外错误率），该错误率可以近似看成交叉验证的结果。

随机森林的优点：1，适用大型数据集，能够在不进行特征缩减的情况下就处理高纬度样本，并且不会过拟合；2，在随机森林构建过程中就能对算法的误差做出无偏估计，给出在分类时变量重要性的排名；3，可以应对大量缺失值存在的数据集，并且在保证一定准确性的情况下估计缺失值。缺点有：1，构建多颗决策树，算法时间复杂度和空间复杂度提升；2，由于随机性的存在，模型减少了可解释性。

本文使用基于CART的随机森林分类器。

2.1.5 AdaBoost

AdaBoost[26]是一种基于boosting提升方法的集成式学习，由Yoav Freund，Robert E Schapire两人根据booting方法改进提出。boosting方法和使用bagging方法的集成式学习大有不同：bagging方法中各个基分类器之间相互独立，没有强依赖关系，并且可以同时训练各个基分类器，基分类器之间是一种并行关系，且最终结果采用集体投票或者平均概率决定。boosting方法的各个基分类器之间具有依赖关系，分类器的训练是串行的，只有训练好一个分类器之后，才能开始训练下一个分类器，是迭代式进行的，且最终结果是由多个弱分类器的输出加权得到的。

AdaBoost算法基于booting方法改进而来，首先，初始迭代时样本权重都是1/N,根据该权重样本集训练一个弱分类器，然后根据当前分类器的结果更新样本权重系数（被错误分类的样本权重增加，正确分类的样本权重减少）。这样在下一次迭代中，算法训练的弱分类器不得不集中于这些被错误分类或者说是上一步中被遗漏的样本。这样经过若干次迭代后就得到了一系列弱分类器，对于每一个弱分类器根据错误率计算权重系数(话语权)，最后将乘以权重系数的各个弱分类器组合得到最终的强分类器。算法具体步骤如下：

1. 样本集为{xi,yi}, yi{1,-1}，i=1,2,…,N。。初始化样本权重D1(i)=1/N。
2. 令k=1,2,…,K,循环k次：
3. 应用带权重的样本集训练出基分类器ck.
4. 计算ck的错误率:

 （2.31）

式中I[]为ck的判别函数，为样本xi在本轮迭代时的权重。

1. 计算ck的权重ak(话语权)：

 （2.32）

1. 更新样本权重系数：

 （2.33）

其中Zk,为规范化因子，此外，根据样本被ck错误分类需要增加其权重，正确分类需要减少权重，还可以应用如下更新公式：

 （2.34）

1. 合并基分类器：对于一个未知样本z，最终分类器CK的判别函数为阶跃函数：

 （2.35）

阶跃函数sign，表达式如式2.36：

 （2.36）

从以上步骤可以出AdaBoost的核心思想就是通过迭代式的训练弱分类器，一步一步的将上一轮迭代中被错误分类的样本提高权重以在下一轮中针对性学习，最后组合弱分类器得到一个强分类器，这样的过程类似于人类学习的过程，每次学习后对于不懂的地方，下一次学习中需要重点对待。

上述AdaBoost算法是针对二元分类问题的，可以轻松的扩展到多分类，并且AdaBoost算法的基分类器可以根据情况选择，目前最常用的分类器是决策树和神经网络模型。

AdaBoost算法的优点：1，算法模型灵活，基分类器选择多样。2，模型精度很高，决策边界平滑，不容易过拟合。3，有扎实的理论支撑，模型结果可理解。缺点：1，由于需要迭代式训练一系列弱分类器，对于大型数据集，时间复杂度和空间复杂度提升。2，对于异常样本较为敏感，可能导致一直分类错误，不断提升权重，最终影响强分类器的稳定性。

本文使用决策树为基分类器的多分类AdaBoost算法。

2.2 深度学习

深度学习是机器学习的子集，深度学习模型即是深度神经网络模型。虽然神经网络模型早在1958年就已经提出（当时叫感知机），但是直到反向传播算法理论发展并应用于神经网络模型，以及计算机硬件发展带来的计算力资源爆炸，神经网络直到2012年由Hinton团队使用深度卷积神经网络模型在ImaginNet机器视觉识别大赛以碾压的姿势赢得冠军，深度学习技术才开始流行起来，并成为现如今所谓人工智能的超级热门领域。而深度学习一词也是Hinton给多层神经网络的学习方法赋予的专有名词。

概括的讲，深度神经网络模型目标是拟合一个目标函数，这个函数可能会超级复杂，含有数量巨大的未知参数。根据通用近似定理，一个具有至少一层隐藏层的前馈神经网络（循环神经网络是一种特殊的前馈神经网络），并且在隐藏层引用的激活函数具有“挤压”性质（如sigmod，ReLU），如果隐藏层的神经单元足够多，就能以任意精度近似任何一个从有限维空间到另一个有限维空间的波莱尔可测函数[35]。因此，具有隐藏层的深度神经网络模型可以看成一个万能函数近似器，如何得到我们想要的模型的关键就是定义模型结构（网络层数目，神经元个数，网络层之间的连接），以及怎么学习得到良好的参数（损失函数，反向传播算法，学习率，学习率衰减等超参数）。

在本节中，将主要介绍卷积神经网络和循环神经网络。

2.2.1 卷积神经网络CNN

卷积神经网络由LeCun于1998年提出[31]，并首先成功应用于手写数字识别。卷积神经网络专门用于处理网格式的数据，最典型的代表就是图像。随着研究者对卷积神经网络的深入研究，以及计算机硬件的发展，卷积神经网络在计算机视觉领域成为绝对优势的深度学习模型。

一个深度卷积神经网络模型通常由若干卷积层、池化层、以及非线性激活函数叠加而成，网络的深度也随着近几年梯度消失和模型恶化问题的解决，加深至1000层以上。下面详细介绍卷积神经网络的各个模块。

1. 卷积：

卷积运算在数学上定义为式2.37：

 （2.37）

形式上是两个函数乘积的积分运算。可以理解成对函数x(a)乘以权重w（t-a）的加权平均。从信号处理的角度理解，x(a)是信号输入，w(t-a)是系统响应，上述运算的结果是系统的输出。从时刻位置理解，x(a)是物体在时刻a的有噪声位置，w(t-a)是以测量结果所在时间a与所求时刻t之间差值为变量（t-a）的权重函数（a越靠近所求时刻，权重越大），s(t)就是对于物体位置平滑估计，是对物体各个时刻求得的位置应用加权平均后的结果。

以上是连续函数的形式。而在卷积神经网络中，输入通常是多维数组，以一张彩色图片为例，输入（input）为3\*H\*W,（H为图片的高，W为图片的宽，单位为像素），一个三维张量，因此对于输入是离散数据形式，需要改造为离散形式：

 （2.38）

将这个函数应用到卷积神经网络中，函数x就是输入，函数w就是核函数（卷积核），输出被称作特征图（feature map）。

对于一个单通道的二维图像，运用一个二维卷积和,卷积运算公式如式2.39：

 （2.39）

在编程实现的时候，许多流行的神经网络库（如pytorch）实现为一个互相关函数：

 （2.40）

如图2.3，演示了如何在二维图像上做卷积运算：



图2.3 二维图像卷积运算示意图

Fig. 2.3 The Schematic diagram of two-dimensional image convolution operation

形象化描述该运算，就如同一个k\*k窗口在图像上滑动，然后对窗口内的值加权平均得到输出特征图上某一个元素的值。这个加权平均的参数和窗口大小就是由卷积核K定义的，卷积神经网络在训练中需要学习的重要参数就是卷积核的参数，通常大小是用户定义的。

卷积运算的重要特性是稀疏连接和参数共享。1，稀疏连接表现在卷积核的大小上。卷积核的大小k通常远小于输入的大小，在相邻两层神经网络中，一个输出单元只受下层k个输入单元的影响，并且处在深层的单元仍然会与浅层的神经单元交互，只是这种交互是间接的。稀疏连接极大的减少了模型运算效率，以及参数的存储需求。2，参数共享，对于卷积核的参数，在输入图中每一位置进行卷积运算时是不变的。卷积核的运算可以理解为这个卷积核固定提取某一特征，比如拉普拉斯运算子的卷积核可以提取图像的边缘特征。这样对一同样类型的特征不管他位于图像任何位置，我们认为它是不变的，因此运用同样的参数就可以提取到该特征。并且，参数共享的特性，还可以应对平移问题，具有平移等变的性质：函数的输入改变，输出也会以同样的方式改变。比如平移一张图像I中的某个对象，那么在输出中对于该对象的表示也会相应平移。更具体的阐述就是，在目标检测中，一只猫的无论它被平移到图像的任何位置，卷积神经网络都能检测到这只猫，只是位置发生了变化。

下述卷积运算可以称为基本卷积运算：不使用0填充（padding=None）,卷积核作用区域移动的步长为1：每次移动一个像素位置（stride=1），设定是否填充，以及步长等可以衍生出其他形式的卷积运算。

假设输入为H\*W大小的图像，使用k\*k的卷积核，步长为s，填充0的个数为p的卷积函数，卷积后输出的特征图大小为h\*w,计算公式为式2.41：

 （2.41）

1. 池化

池化的目的是减少参数，并使得输入的表示近似不变。池化函数用某一区域内输出的某种统计特征（最大，平均，加权平均）代替网络在该位置的输出，如图2.4所示最大池化函数（max pooling）用2\*2滑窗矩形区域的最大值作为该区域的输出：



图2.4 最大池化示意图

Fig. 2.4 The Schematic diagram of max pooling

另一种最常用的池化函数是平均池化，取滑窗区域内的平均值作为输出。池化函数其实也可以理解为一种特殊的卷积运算，它的卷积核没有参数，只有操作（最大，平均），步长stride=2（一般跟卷积核k大小一样）。因此采用2.41式可以计算出应用卷积核为2\*2，步长为2的最大池化函数后输出大小为输入的1/2：

 （2.42）

池化操作让卷积神经网络产生了平移不变性。在最大池化中，如果输入图像进行了少量平移，由于最大池化每次取池化空间区域的最大值，即使图像平移，那个最大值也会相应平移，但是仍会被激活作为输出值。由于池化是一个区域内的综合统计特征，使得池化后输出单元能够代表输入单元，因此极大的减少了输入大小，提高了计算效率与存储效率。

1. 全连接层

全连接层是输入和输出之间使用一个大的权重矩阵将输入映射到输出：

 （2.43）

其中，W是权重矩阵，X是输入矩阵，b是偏置。全连接层的特点就是单元和输出单元均有连接（这也是全连接层的由来），权重矩阵只使用一次，没有参数共享。全连接层的作用是聚合前面卷积和池化组成的骨干网络输出的特征图的全局信息，然后映射到输出空间。当然，全连接层并不是卷积神经网络所特有的，其实全连接层是神经网络的最基础结构。

综上，卷积神经网络的组成由两部分，一部分是卷积层和池化层组成的骨干网络，它用于从输入中提取抽象的特征，一部分就是全连接网络，用于聚合全局特征然后映射到输出空间，完成任务预测。这是最基础的卷积结构，后续研究者，开发和改进了更加优秀且复杂的结构，几个最重要的结构改进有深度可分离卷积，Dropout机制，残差网络的跳跃连接。跳跃连接解决了超深层（操过100层）卷积神经网络的梯度消失和模型恶化问题，并且高效复用了特征。

2.2.2 循环神经网络RNN

循环神经网络是为了专门处理序列格式数据的一类神经网络。循环神经网络在每个时间步上应用相同的参数，通过参数共享的方式来记忆和提取信息。在最常用的机器翻译应用场景中，输入数据文字就是一种序列化数据，序列中的元素是一个个单词。我们在翻译的时候，需要联系上下单词，考虑到单词之间的长距离依赖关系，因此需要记忆功能，需要容忍单词出现顺序的变动（单词位置变了，单句子意思没变），这些都可以用循环神经网络来解决。

对于输入序列（x(1), x(2), x(3),…, x(t)）,每一个时间步应用同样的更新方程，则循环神经网络的前向传播公式如图：

 （2.44）

 or  （2.45）

 （2.46）

 （2.47）

式中，f代表激活函数，可以取Tanh或者ReLU。**h**(t)是时间步为t时的隐藏层状态向量，**W**是上一个隐藏层输出到下一个隐藏的连接的参数矩阵，**U**是输入到隐藏的连接参数化矩阵，**V**是隐藏到输出的参数化矩阵，**c**，**b**分别是偏置向量。典型的循环神经网络结构如图2.5:



图2.5 循环神经网络示意图

Fig. 2.5 The Schematic diagram of of recurrent neural network

由图2.5所示，每一个时间步后隐藏层状态向量**h**(t)都编码了前面所有序列的信息。在每个时间步都可以根据**h**(t)产生一个输出**y**(t),使用这种方式能够生成一个和输入序列等长的输出序列，只能映射等长序列到等长序列，使得引用非常局限，而且也不符合语言翻译的之间两种语言序列长度通常不同的现实。在最后一个时间步后，**h**(t)事实上是编码了样本整个序列的信息，那么是否可以直接应用这个上下文向量**h**(t)解码得到另一个序列呢？答案是肯定的，使用RNN作为解码器，研究者提出了基于编码-解码的序列到序列架构。

上述循环神经网络的结构不能处理非常长的长期依赖问题，只能处理较为简单的序列建模问题，为了解决长期依赖的问题，研究者们提出了门控RNN(gated RNN):包括长短期记忆网络（long short-term memory，LSTM）和基于门控循环单元（gated recurrent unit,GRU）的网络，这两种网络通过门来决定继续积累状态信息，还是选择遗忘积累的信息，这些选择由各个门的权重自动决定，并且在每个时间步都会动态改变。

1. 长短期记忆网络LSTM



图2.6 LSTM示意图

Fig. 2.6 The Schematic diagram of of LSTM network

如图2.6所示的LSTM结构图[23]，LSTM有三种门，遗忘门ft ,输入门it ,输出门ot，这几种门均被sigmoid函数将权重设置在0到1之间。有两种状态，隐藏状态ht,单元内部状态Ct。根据图2.5，LSTM在第t个时间步时， 各个门和状态计算公式如下式2.48：

 （2.48）

在式2.45中，遗忘门控制存储在上一时间步状态Ct-1中的信息以多大程度被遗忘，取1为不遗忘，让信息通过，取0则为全忘掉。

 （2.49）

 （2.50）

在式2.49中，输入门it由sigmoid函数决定将什么新输入更新到内部状态Ct中，式2.50根据xt和ht-1生成候选内部状态。

 （2.51）

式2.51表示如何更新内部状态Ct,表示逐元素相乘。可以看到，新的内部状态由过去的状态点乘遗忘门加上新的候选状态点乘更新门得到。新的状态Ct需要根据当前输入决定遗忘过去并记忆现在。

 （2.52）

 （2.53）

在式2.52给出了输出门ot的计算公式，门ot被sigmoid函数控制权重在0-1之间，控制输入信息通过该门的信息量的大小。隐藏状态ht由输出门点乘经过tanh函数的当前内部状态Ct决定，如式2.53。综上所述，LSTM与传统循环神经网络在大的框架上一致，均是由输入xt和ht-1来不断计算新的h，在计算的过程中采用相同的更新公式，进行参数共享。不同的是LSTM在内部结构加入各种门来自动控制信息的传递，以解决长期依赖。

2）门控循环单元GRU

LSTM计算较为复杂，有些部分经过研究认为可以简化，2014年GRU由Cho等第一次提出[32]。GRU将门进行简化，让单个门控单元“更新门”同时控制“遗忘门”和“输入门”的决定，计算公式如下式2.54：

 （2.54）

其中，更新门zt，复位门rt，以及计算公式如下：

 （2.55）

 (2.56）

 (2.57）

可以看到，GRU的更新计算中，去除了内部状态c，并将三个门简化为两个门：更新门和复位门。更新门可以选择完全复制过去状态(ht-1)而忽略当前输入（zt=1）也可以由当前新的状态（）替换掉过去的状态（zt=0），当然这都是两个极端，大部分情况是中间某个值，输出的新状态是过去和当前输入的结合。复位门控制过去状态ht-1中多少部分用于生成下一个状态ht。

2.3 本章小结

本章内容为异常入侵检测相关技术研究。主要阐述了当前异常入侵检测技术所使用的技术，主要为两类，一类是传统机器学习算法，以人工特征数据集为基础，介绍了如决策树，支持向量机，贝叶斯，随机森林，AdaBoost这5种经典算法，并分析了优缺点。另一类是深度学习算法，也是本文研究的重点，在这一小节种又详细介绍了深度学习算法中目前应用在异常入侵检测中较多的卷积神经网络和循环神经网络。深度学习算法以强大的表示学习能力能够自动从数据中提取高级抽象特征，成为目前异常入侵检测算法的热点，且性能在大型数据集上表现优秀，是本文选择深度学习算法的原因。

1. 基于CICIDS-2018的会话样本数据集构建

作为以数据驱动的深度学习算法，数据会很大程度会影响算法性能。在第2章的技术介绍中，深度学习方法具有从原始网络流量数据中学习并提取高级抽象特征的能力，但是使用该方法面临着一个困难，就是数据的获取和处理，合理规划样本数据包字节，和数据包数，从而构建一个可直接用于深度学习的较为全面的数据集。本文没有自己捕获网络流量数据，而是使用CICIDS-2018提供的网络流量数据pcap文件。在该pcap文件的基础上，本文在统计pcap文件中数据包大小分布，时间分布等信息上，提出了一种流量分割算法，将pcap文件分割成以会话为独立存在的一个个样本（最后结果为张量矩阵），设定了样本中会话的最大数据包数目，会话中每个数据包的最大字节数等细节，最后构建了一个包含多种攻击类型，可直接用于深度学习算法的会话样本集。

## 3.1 数据集构建方法

构建步骤如下：

1.通过对CICIDS-2018的流量数据进行分许，确定数据集单个会话样本的具体结构。主要为两个参数，单个数据包的最大字节数，样本包含的最大数据包数。

2.选取合适的pcap文件。网络流量文件pcap提供的是整个网络的所有主机的流量文件，选取合适的主机的pcap文件进行分割提取，主要是被攻击机器所在的文件。

3.详细介绍流量分割方法流程。

4.编写流量分割算法的代码，输入网络流量pcap文件，将所有的样本以矩阵形式存储。

5.为每个样本打上标签，根据协议，ip，端口，时间等信息将每个样本人工标记标签，需要格外细致，构建结束。

3.1.1 CICIDS-2018网络流量数据最大数据包字节数，最大数据包个数

CICIDS-2018数据集是当前较新的入侵检测数据集。它通过统计学方法封装用户的正常网络行为产生的流量的网络特征，这些特征包括各种网络协议数据包的尺寸，数据包数目，请求时间分布等。封装的用户行为用于生成网络中的良性事件，通过这种行为产生的网络流量可以看成是真实的流量。使用明确描述的攻击场景在一个LAN网络拓扑结构中实施攻击。如下图3.1，实施了十天的攻击，每天的攻击包含不同的攻击内容，最后收集网络中的流量，记录为pcap捕获文件。

Pcap文件是不能直接使用的，需要进行处理，现在有两个方向，一个是使用流量分析工具，提取基于统计学和专家知识的各种流量特征，该数据集的构建者就提供了基于这10天捕获流量的统计分析特征，它们使用一个叫CICFlowMeter的流量分析工具生成了包含共计80个流量特征的特征数据集，数据集中每个样本是一个双向流（会话），共十个文件，我们对下载下来的10个文件进行分析，和一些基本的数据清理（nan值填0），最后选用如下8个文件，如表3.1：

表3.1 CICIDS-2018网络流量数据攻击情况表

Table3.1 CICIDS-2018 network traffic data attack situation table

|  |  |  |
| --- | --- | --- |
| **序号** | **捕获文件** | **包含攻击** |
| 1 | Wed-14-02-2018 | FTP-BruteForce, SSH-BruteForce |
| 2 | Thurs-15-02-2018 | Dos-GoldenEye, Dos-Slowloris |
| 3 | Friday-16-02-2018 | Dos-SlowHTTPTest, Dos-Hulk |
| 4 | Tues-20-02-2018 | DDoS-LOIC-HTTP |
| 5 | Wed-21-02-2018 | DDoS-LOIC-UDP, DDoS-HOIC |
| 6 | Thurs-22-02-2018 | BruteForce-Web, BruteForce-xss, SQL-injection |
| 7 | Wed-28-02-2018+ Thurs-01-03-2018 | Infiltration |
| 8 | Friday-02-03-2018 | Bot |

官方提供的csv文件中有一些错漏，进行了一些改进：1，Tues-20-02-2018文件中在官方的说明中应包含两种攻击：DDoS-LOIC-HTTP和DDoS-LOIC-UDP,但其实只包含一种攻击DDoS-LOIC-HTTP，并且该文件包含近800万条数据，前100w条就包含了所有的攻击样本数据，余下都是正常样本，因此对该天csv文件处理，取前100万个样本数据即可；2，在序号3，7文件中包含重复的序列名，且出现在样本列中，如果不进行数据清洗，删除这些数据，会造成读取错误。3，在实行渗透攻击的两天文件中（Wed-28-02-2018+ Thurs-01-03-2018）总共只包含100多万条数据，进行合并处理。4，对于web端攻击只选取了一天的数据Thurs-22-02-2018。

从表3.1可以看出，ids2018包含的攻击有6种：暴力破解，Dos拒绝服务，DDoS分布式拒绝服务，web端攻击，渗透，僵尸网络。细分的话包含14种攻击小类。上述总共包含800多万个样本。

3.1.2 最大数据包字节数，最大数据包个数

在本小节中，我们需要对捕获文件进行一定分析，从而确定两个关键值：1，数据包字节数：选取的数据包字节数过大易造成存储资源，计算资源浪费，过小则不能得到有效信息，影响算法准确度。2，数据包数，同理，过大过小都不合理，需要合理估计。对于这两个参数，本文从统计数据的角度进行分析确定。

1. 单个数据包可存储最大字节数

对于每一天捕获的文件，我们通过CICIDS官方提供的对应的人工特征集，特征中有单个样本的字节数，数据包数等，对其进行统计分析。如表3.2，为10个文件的关于数据包字节的统计数据，单位为字节byte。

表3.2 IDS-2018数据包字节统计参数表

Table3.2 IDS-2 network traffic data attack situation table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **文件** | **样本数** | **数据包最大字节** | **最大平均字节** | **全部平均字节75%** |
| 1 | Wed-14-02-2018 | 1048575 | 64440 | 332 | 122 |
| 2 | Thurs-15-02-2018 | 1048575 | 64440 | 330 | 147 |
| 3 | Friday-16-02-2018 | 1000000 | 8688 | 402 | 187 |
| 4 | Tues-20-02-2018 | 1048575 | 1460 | 438 | 241 |
| 5 | Wed-21-02-2018 | 1048575 | 2224 | 366 | 187 |
| 6 | Thurs-22-02-2018 | 1048575 | 40120 | 360 | 169 |
| 7 | Wed-28-02-2018+ Thurs-01-03-2018 | 944171 | 10024 | 331 | 150 |
| 8 | Friday-02-03-2018 | 1048575 | 1711 | 210 | 163 |

首先，样本是一个双向流，流中包含一系列数据包。在表中，最大平均计算的是流中最大数据包尺寸的平均值，全部平均，是先计算流中数据包尺寸的平均值，在计算该平均值在真个样本集中统计平均值。75%分位点是指在该点数据值之下包含近75%的样本。

分析可以得出，整个样本集中单个数据包的字节数变化范围极大，从只包含的协议数据头的最小字节到ip最大字节数都有分布。但从全部平均75%分位点可以看到，平均字节大概在200个字节左右。从最大平均字节而言，最大数据包字节数都在300左右。因此，从尽量将完整数据包存储进样本矩阵中角度看，选取数据包字节数：**320**较为合理。

1. **单个样本最大包含**数据包个数

一个样本表示为一个双向流，为两个机器之间通信的所有的数据包。在真实的网络中，会话中的数据包个数是非常不确定的，但是一般认为其符合统计规律，为正态分布，因此选取一个较为合理的值就可包含绝大部分数据。

表3.3 CICIDS-2018样本中数据包个数统计参数表

Table3.3 CICIDS-2018 network traffic data attack situation table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **文件** | **样本数** | **最大数据包个数** | **平均数据包个数** | **75%分位点** |
| 1 | Wed-14-02-2018 | 1048575 | 9198 | 13 | 13 |
| 2 | Thurs-15-02-2018 | 1048575 | 19181 | 11 | 9 |
| 3 | Friday-16-02-2018 | 1000000 | 889 | 5 | 8 |
| 4 | Tues-20-02-2018 | 1048575 | 280043 | 98 | 7 |
| 5 | Wed-21-02-2018 | 1048575 | 309629 | 197 | 8 |
| 6 | Thurs-22-02-2018 | 1048575 | 38418 | 15 | 15 |
| 7 | Wed-28-02-2018+ Thurs-01-03-2018 | 944171 | 121309 | 11 | 11 |
| 8 | Friday-02-03-2018 | 1048575 | 69241 | 13 | 12 |

从表3.3可以看出，单个样本中（双向流）含有的数据包个数数目相差极大，但是从75%分位点有可以看出绝大部分样本数据包含的数据包个数不超过20。为了尽可能的将会话样本中的数据包都存储进矩阵样本中，选取一个较大的最大数据包数：**32**。

综上，确定了接下来流量分割中重要的两个参数，一个是单个数据包最大字节（表示了对会话中单个数据包信息的保留程度），一个是单个会话样本能包含的最大数据包个数（表示了单个样本能保留原始会话样本总体信息的能力），两者的确定原则是尽可能保留原始字节信息，也要避免选取参数过大，造成计算资源增加，以及模型低效而冗余。

3.1.3 流量分割方法流程

首先，我们按照如下两个五元组定义一个会话：（源ip地址，目的ip地址，协议，源端口，目的端口），（目的ip地址，源ip地址，协议，目的端口，源端口）。会话样本就是两个机器之间通信的所有数据包的集合。流量分割的目的就是将pcap文件以会话为单位进行分割，并将原始数据包字节以矩阵形式进行存储。

如图3.1所示，为流量分割流程示意图。

图3.1 流量分割流程图

Fig. 3.1 The Flow Chart of Traffic Segmentation

图3.1描述了本文流量分割方法的流程。首先，从pcap文件读取一个数据包，根据数据包的协议信息，提取5元组。并且提取该数据包的原始字节信息，每8bit为一字节，可以采用张量的形式进行存储，如numpy文件或者torch形式的张量文件（按照之前确定的参数截取该数据包的320个字节，超出的字节截断，少于320则填0至320）。在进行以上步骤后，根据该数据包的五元组，应该在五元组列表里进行遍历（从尾部开始，即从最近时间的会话样本开始）：如果寻找到了同样的五元组，则进入超时判断。对于超时，一般是服务器定义的，对于CICIDS-2018数据的构建者，官方取了600s为超时时间（对于该参数，可以根据现实的服务器定义进行设置），对于超时会话，应该按照新的会话进行处理，非超时会话，则将该数据包进行320字节截取加入已有的会话矩阵中（例如，该会话已存储了7个数据包，则该矩阵大小为7\*320，加入一个数据包后就变为8\*320）；没有寻到相同五元组，按照新建会话的流程进行处理（新建矩阵文件，将五元组加入记录列表）。最后，遍历该pcap文件所有数据包至完成流量分割。

本方法的流程不局限于单个pcap文件，可以将多个pcap文件联合进行处理。更多细节见算法3.1所示：

Code 3.1 流量分割算法

Require:捕获文件pcap

Require:超参数，数据包最大字数320，会话含有的数据包最大个数32

1. 新建五元组记录列表flows[]

2. 新建记录会话第一个数据包捕获时间列表times[]

3. 初始化session\_id=0

4. While pcap文件没有读取完毕 do：

5. 读取数据包packet

6. 从packet获取flow=（源ip，目的ip，协议，源端口，目的端口）

7. 读取packet.time,得到该包捕获时间time

8. 获取320个字节，得到一个向量a

9. for 从flows最后一个记录开始遍历do:

10. if 五元组和flows以有记录相同（正向，反向有一个即可）：

11. if 没有超时：

12. 32以内：读取相应会话存储矩阵文件，与a进行拼接

13. 32以外：记录真实数据包数

14. break

15. else超时：

16. a.新建会话，a存储为矩阵文件，文件名为session\_id.pt

17. b.将五元组加入记录列表，flows.append(flow)

18. c.记录会话第一个包的时间，times.append(time)

19. d.session\_di += 1

20. break

21. if flows列表遍历完毕也没有找到相同会话：

22. 重复步骤a，b，c，d

23. end while

24. 将flows存储成csv文件，times作为一列也加入该scv文件

算法3.1展示了流量分割中的主体代码逻辑。经过这一系列步骤，我们得到了一个csv文件，这个文件标识了分割中的每一个会话，并且通过该scv文件我们通过对应的session\_id就可以轻松找到矩阵文件session\_id.pt。此外为了之后的逻辑分析，我们还可以在流量分割中计算一些有意思的参数统计，如会话的数据包真实值（由于最大32个数据包的限制，矩阵文件size无法反应真实值），计算会话持续的时间（最后一个数据包的时间减去第一个数据包的时间），数据包真实尺寸的平均大小，以及tcp会话完整程度等参数，作为新的一列加入csv文件中。

3.2 会话样本数据集的构建过程

3.2.1 准备工作

CICIDS-2018是在云计算平台上的一个LAN网络拓扑上实施收集的。网络拓扑如图3.2所示：



图3.2 ids-2018的网络拓扑图

Fig. 3.2 The Network Topology of CICIDS-2018

在表3.1中展示了每天实施的不同类型的攻击，在实验进行流量分割时，每个攻击场景都会有特定的受害者机器。因此不需要使用全网的LAN捕获流量文件，只需要找到受害者机器的当天捕获流量文件进行提取即可。在实际流量分割中，计划在每天所实施的攻击场景中提取10万个会话样本。

表3.3展示了提取的攻击和pcap文件所属关系：

表3.3 用于流量分割的pcap文件

Table3.3 the pacp file for traffic segmentation

|  |  |  |
| --- | --- | --- |
| **序号** | **日期** | **用于流量分割的pcap文件名** |
| 1 | Wed-14-02-2018 | UCAP172.31.69.25  UCAP172.31.69.28 |
| 2 | Thurs-15-02-2018 | UCAP172.31.69.25 |
| 3 | Friday-16-02-2018 | UCAP172.31.69.25-part1.pcap  UCAP172.31.69.25-part2.pcap |
| 4 | Tues-20-02-2018 | UCAP172.31.69.25 |
| 5 | Wed-21-02-2018 | UCAP172.31.69.28-part1  UCAP172.31.69.28-part2 |
| 6 | Thurs-22-02-2018 | UCAP172.31.69.28 |
| 7 | Wed-28-02-2018 | capEC2AMAZ-O4EL3NG172.31.69.24-part2 |
| 8 | Friday-02-03-2018 | capEC2AMAZ-O4EL3NG-172.31.69.23  capEC2AMAZ-O4EL3NG-172.31.69.17  capEC2AMAZ-O4EL3NG-172.31.69.8  capEC2AMAZ-O4EL3NG-172.31.69.12  capEC2AMAZ-O4EL3NG-172.31.69.29  capEC2AMAZ-O4EL3NG-172.31.69.30 |

由于被受到攻击的机器的流量含有大量攻击流量，为了类别平衡，在提取足够样本的攻击样本后，可以选取同一部门的其他主机的流量pcap文件提取正常样本。

3.2.2 分割结果

现在流量分割的所有准备工作都已做完，可以正式进行流量分割，运行算法，输入文件，最后分割结果如下：

表3.4 流量分割样本结果

Table3.4 the Traffic segmentation sample results

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **日期** | **攻击类别和数目** | **正常样本数目** |
| 1 | Wed-14-02-2018 | FTP-BruteForce：29599  SSH-Bruteforce：36510 | 33891 |
| 2 | Thurs-15-02-2018 | DoS-GoldenEye：25323  DoS-Slowloris：6979 | 67698 |
| 3 | Friday-16-02-2018 | DoS-SlowHTTPTest：14116  DoS-Hulk：14116 | 71768 |
| 4 | Tues-20-02-2018 | DDoS LOIC-HTTP：49703 | 50297 |
| 5 | Wed-21-02-2018 | DDoS-LOIC-UDP：110  DDOS-HOIC：99727 | 163 |
| 6 | Thurs-22-02-2018 | Brute Force -Web：137  Brute Force -XSS：43  SQL Injection：17 | 99803 |
| 7 | Wed-28-02-2018 | Infiltration:88144 | 11856 |
| 8 | Friday-02-03-2018 | Bot:46571 | 54462，53430 |
| 总计 | 80万会话样本 | 411102 | 388898 |

如表3.4所示，本次流量分割实验对8天包含14种攻击流量的pcap文件进行分割。共计得到80万个会话样本，其中异常会话样本有411102个，正常会话样本388898个，比例上正常样本和异常样本基本平衡。在序号6所在的流量分割实验中，异常样本极少，类别较为不平衡。

3.2.3 样本标签

分割出样本后需要给样本打上标签，根据CICIDS-2018提供的攻击场景实施表可以给出标签判断的依据。在流量分割实验中，得到的csv文件中我们对每一个样本都会有一条对应的记录，包括五元组，数据包数等一些统计数据，这些记录和存储在会话样本矩阵文件是分离的，根据记录可以逐个给每个样本打上标签。

样本标签的流程如下：实施攻击的时间匹配，使用的协议匹配，端口匹配（是常用协议的端口如80，21），ip地址匹配（攻击者或被攻击者ip地址，根据情况，比如ddos就不止几个攻击者ip）。这几个匹配条件根据攻击类型有所变化，在这几个条件都符合后，就可以判断该会话样本是异常样本。

在本论文中，分配的标签如下： {'Benign': 0, 'FTP-BruteForce': 1, 'SSH-Bruteforce': 2, 'DoS attacks-GoldenEye': 3, 'DoS attacks-Slowloris': 4, 'DoS attacks-Hulk': 5, 'DoS attacks-SlowHTTPTest': 6, 'DDoS attacks-LOIC-HTTP': 7, 'DDOS attack-HOIC': 9, 'DDOS attack-LOIC-UDP': 8, 'Brute Force -Web': 10, 'Brute Force -XSS': 11, 'SQL Injection': 12, 'Infilteration': 13, 'Bot': 14}。

3.3 会话样本数据集分析

3.3.1 可视化分析

将存储的会话样本表示为二维图像，按照如下规则融合数据包：先生成一个0填充的32\*32共计1024个元素的矩阵。取出会话样本中的数据包，只取数据包中的有效字节（在之前流量分割中样本存储策略，每个数据包截取320个字节，多余的截断，不足的填0，这里只需要取非0填充的字节），将这些数据依照从上到下，从左到右的顺序填充到32\*32矩阵中。取下一个数据包的有效字节在上一个数据包的下一个位置顺序填充，直到32\*32矩阵填充完毕，或者数据包取完。这样，矩阵中每一个元素就是一个字节，该矩阵就是一个32\*32的二维黑白图像，最多容纳1024个字节。

图3.3为通过FTP,SSH进行暴力破解的异常会话样本可视化图像。FTP-BruteForce异常会话包含2个数据包，因此可以看到图像下半部分为黑色。SSH-BruteForce异常会话包含44-46个数据包，图像呈现一定规律性。

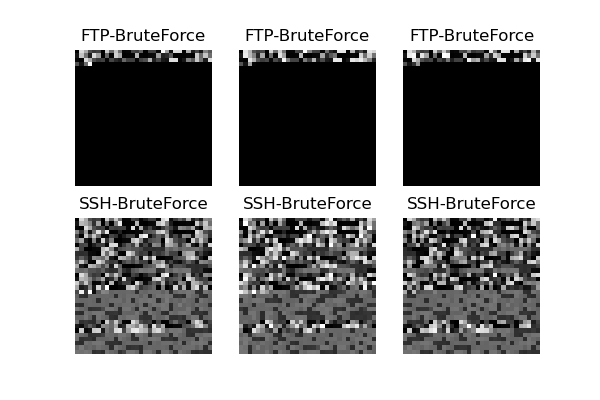


图3.3 FTP,SSH -BruteForce可视化示意图

Fig. 3.3 The Visualization diagram of FTP, SSH -BruteForce

图3.4 呈现了两种拒绝服务攻击的可视化图像。Dos-GoldenEyeyi异常会话样本由8个数据包，Dos-Slowloris异常会话有18个数据包和3个数据包两种情况，图3.4最下两行分别依次展示了18个数据包，3数据包的可视化图像。

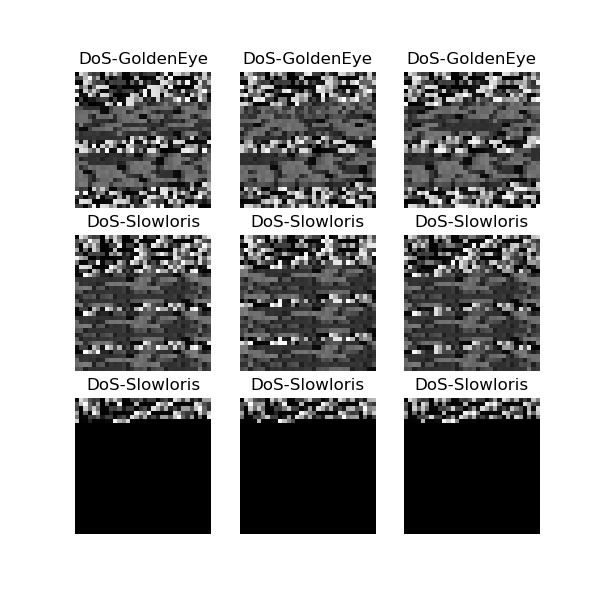


图3.4 Dos-GoldenEye,Slowloris可视化示意图

Fig. 3.4 The Visualization diagram of Dos-GoldenEye, Slowloris

图3.5是SlowHTTPTest,Hulk两种拒绝服务攻击类型的可视化图像。SlowHTTPTest的异常会话样本包含数据包数为114-16，Hulk会话为60左右。

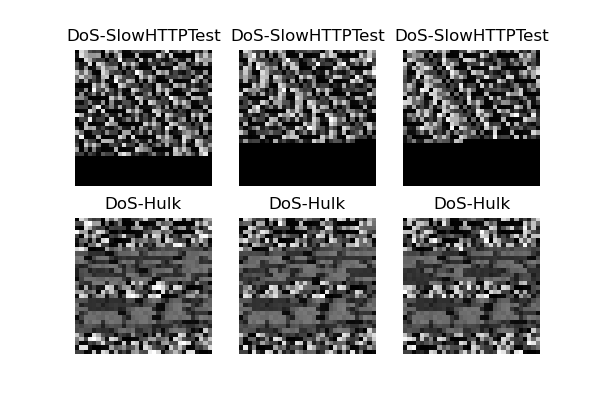


图3.5 Dos-SlowHTTPTest,Hulk可视化示意图

Fig. 3.5 The Visualization diagram of Dos-SlowHTTPTest and Dos-Hulk

图3.6 呈现了3种分布式拒绝服务攻击会话样本的可视化图像。

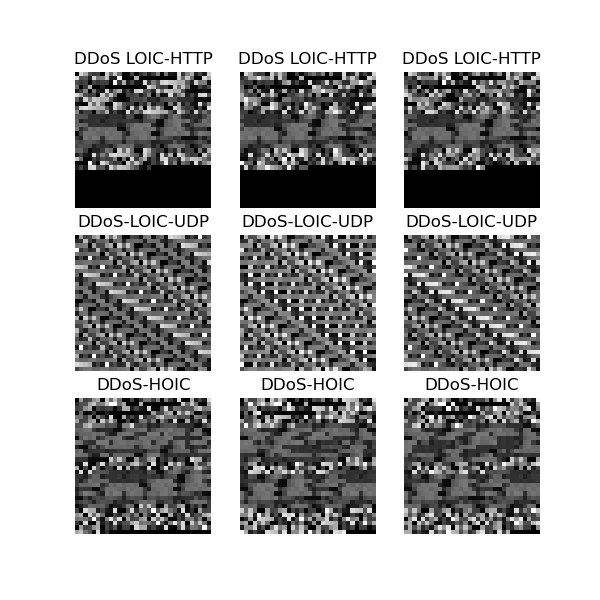


图3.6 DDoS可视化示意图

Fig. 3.6 The Visualization diagram of DDoS

LOIC是低轨道离子加农炮的缩写（the Low Orbit Ion Cannon），是一种发动分布式拒绝服务的工具。LOIC-HTTP异常会话一般有9个数据包，这种攻击的策略是建立大量的http连接来消耗服务器的资源。LOIC-UDP一般包含多达百万个udp数据包，是典型的洪泛攻击，消耗服务器的带宽资源，使其不能响应正常用户的数据。HOIC是高轨道离子加农炮的缩写（the High Orbit Ion Cannon），与低轨道的区别在于同时可以对多个服务器url资源发动攻击，该攻击异常会话样本包含10个数据包。

图3.7 为web端攻击会话样本可视化图像。其中两种暴力破解会话样本包含较多数据包（为260，310），而SQL注入较少，只有10个左右数据包。

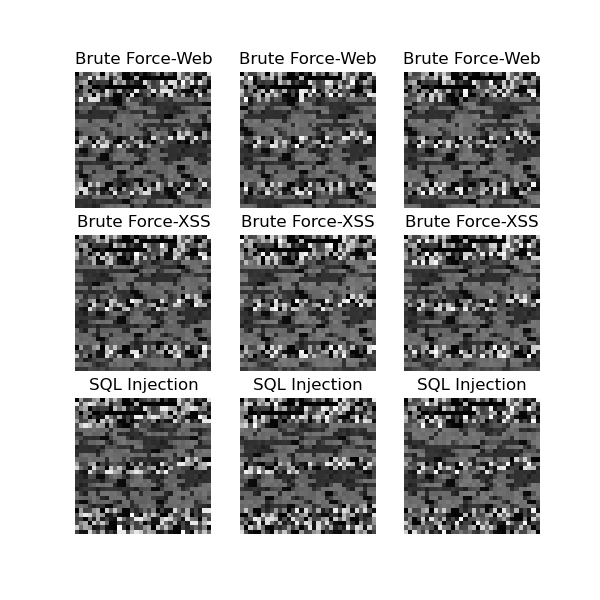


图3.7 BruteForce(web xss) and SQL Injection可视化示意图

Fig. 3.7 The Visualization diagram of Brute Force (web xss) and SQL Injection

图3.8 是渗透攻击（infiltration），僵尸网络攻击（Bot）的可视化图像。渗透攻击通过钓鱼邮件，在目标机器里植入后门程序，然后就可以进行渗透攻击，利用Nmap扫描内网的ip，使用的端口，打开的服务等，获取内部网络信息。因此渗透攻击通常表现为受害主机对内网的机器进行频繁而简单的通信，该异常样本数据包数目通常为1-3个。僵尸网络攻击则是对受到感染植入了Bot客户端程序的异常通信行为，如定时与僵尸网络服务器通信，发送受害者机器的信息，如浏览器记录，键盘按键，屏幕截图等，该异常会话样本数据包数为10个。

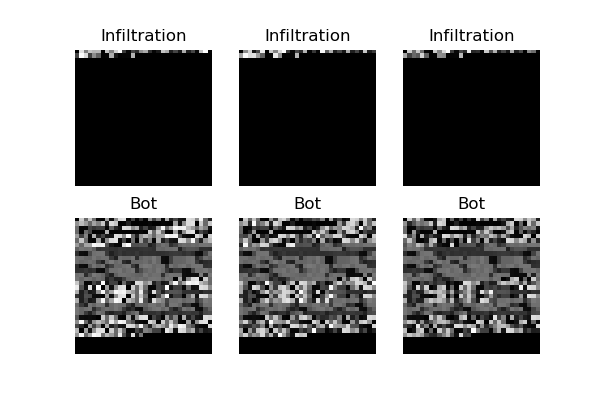


图3.8 Infiltration,Bot可视化示意图

Fig. 3.8 The Visualization diagram of Infiltration and Bot

通过以上可视化图像分析，我们从图中可以看到明显的图形特点，各种异常会话样本具有一定的规律性。这些特定的图形规律将使其区别于正常通信的会话可视化图形，因此利用深度学习提取会话样本的二维图像特征来建模网络行为具有可行性。

3.3.2 数据包数目，数据包字节，会话持续时间

我们想要了解我们构建的数据集的样本中数据包数目的分布情况。数据包数目是离散的而且有极端的点，为此，进行一个简单的处理，将数据包数目大于50的数据包记为50，以约束分布的边界，对于数据包分布的极端值，我们可以从数据的最大值获得。下表3.5为4个统计量的统计分析表，其中packets为单个会话样本中数据包数目，bytes为会话总字节数，mean\_byte为总字节除以数据包数后的平均字节数，duration为会话的持续时间。

表3.5 会话样本统计分析表

Table3.5 Session sample statistical analysis table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **最小** | **平均** | **50%** | **75%** | **最大** |
| Packets (个) | 1 | 3 | 9 | 10 | 2542059 |
| Bytes (byte) | 42 | 2820 | 1019 | 1841 | 1.8\*10^8 |
| Mean\_bytes(byte) | 15 | 146 | 120 | 181 | 23992 |
| Duration (秒) | 0.00001 | 487 | 0.037850 | 5.269004 | 40681.466220 |

核密度估计的方法可以估计数据潜在的概率密度函数，其类似于包裹直方图的连续平滑曲线。使用高斯核函数进行连续的密度估计，可以绘制密度分布图，横坐标为数据值，纵坐标为该店值得概率。根据密度分布图可以绘制经验累积分布图，横坐标为数据值，纵坐标为在所有小于等于该值的样本占所有样本的比例。如下图3.9为数据包数目packets的密度分布图和经验累积分布图。

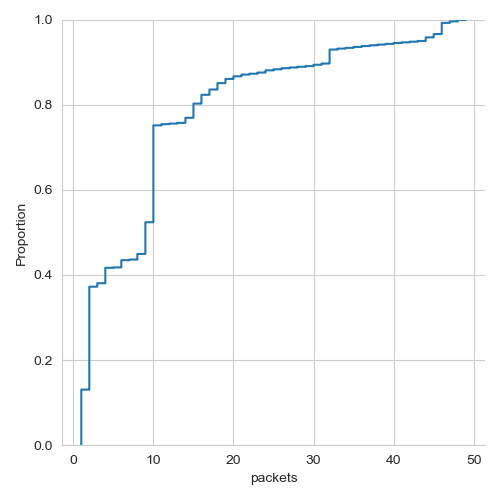
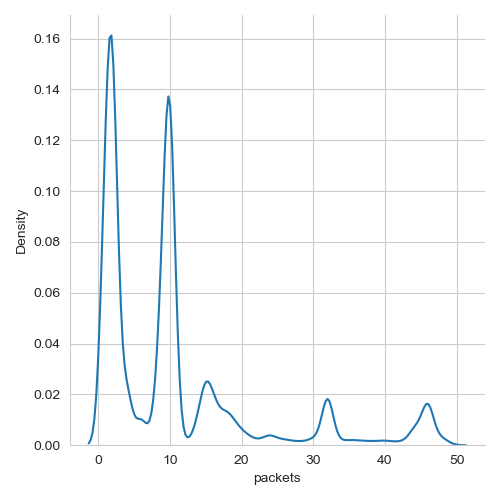


图3.9 会话数据包个数，密度分布图，经验累积分布图

Fig. 3.9 Density distribution and Experience cumulative distribution of packets

从图3.9我们可以看到，数据包数目分布最多的值范围是0-50左右，而在数据包个数为32时，从经验累积分布图可以看到90%的样本包含的数据包都不超过32个。

图3.10展示了样本总字节数的分布情况。密度分布图的几个高峰是54，3000，7500左右。从累积分布图看出95%的样本字节数在9000以下。

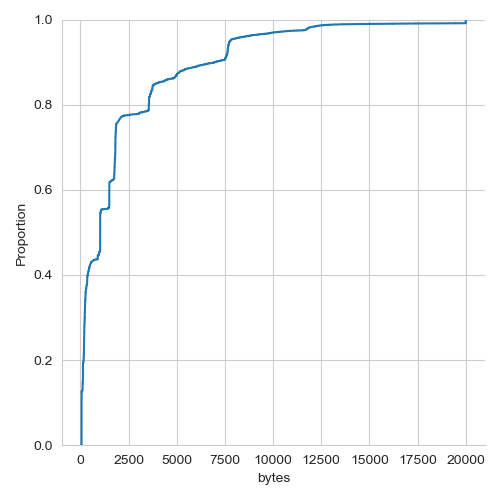
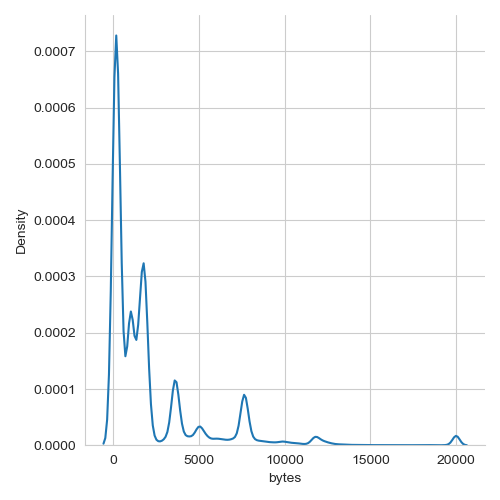


图3.10 会话总字节，密度分布图，经验累积分布图

Fig. 3.10 Density distribution and Experience cumulative distribution of bytes

图3.11是会话样本中的单个数据包的字节分布图。从密度分布图可以看到，平均字节分布最多的几个值是，54，108，174左右，从密度分布图看到，近乎98%的样本中单个数据包含有的字节不超过400，95%的样本中单个数据包的自己不超过300个。

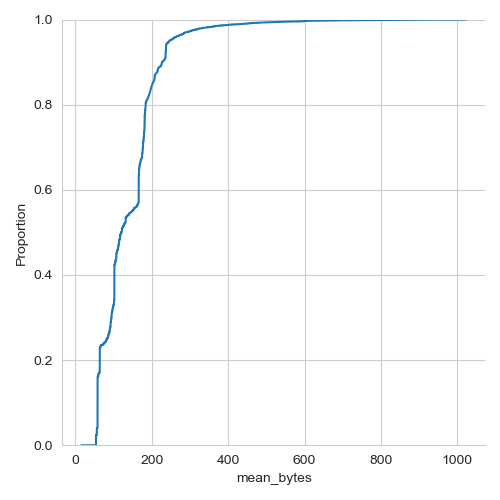
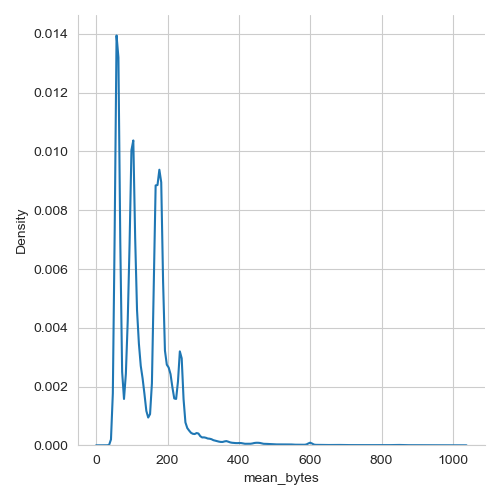


图3.11 会话平均字节，密度分布图，经验累积分布图

Fig. 3.11 Density distribution and Experience cumulative distribution of mean\_bytes

图3.12 展示了会话持续时间的分布情况。从密度分布图来看，会话持续时间分布的两个峰值范围为1秒，与1000秒以上两个区域。从经验累积分布图看到，近80%的会话持续时间都不超过20秒。相对极端是类似于LOIC-UDP这种会话，持续时间为2000秒左右，并且在这段时间发送巨量的数据包来消耗服务器的资源。

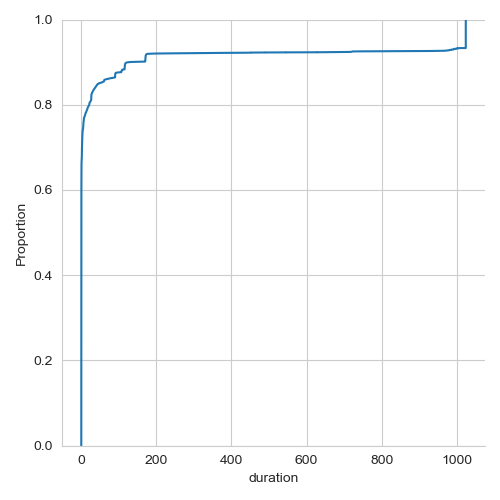
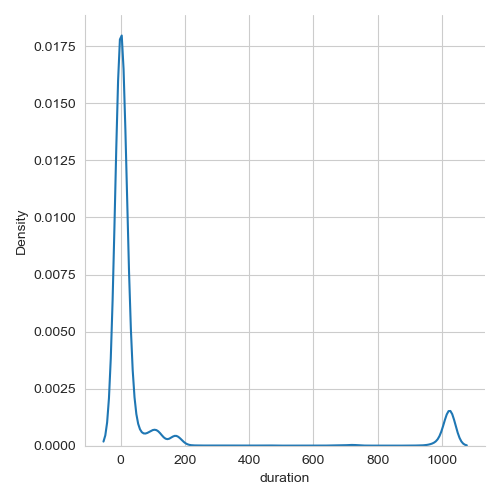


图3.12 会话持续时间，（左）密度分布图，（右）经验累积分布图

Fig. 3.12 Density distribution and Experience cumulative distribution of duration

本节对基于CICIDS-2018提供的原始网络捕获文件，通过流量分割的方法得到会话样本数据集进行了分析。每个样本是一个会话，会话中含有一系列数据包，每个数据包最多截取320个字节，每个会话最多保留32个数据包，这样80万个样本形成的数据集进行了统计分析。分析的内容是：每个会话真实的数据包个数的分布，会话字节分布，会话持续时间分布等。这些数据首先验证了之前流量切割方法中两个限制（320字节，32个数据包）设置的合理性，同时也对网络流量的一些统计特征有了一定认识，提供了可视化的图像供研究者参考。

3.4 本章小结

本章主要内容是构建一个可用于深度学习的基于网络流量数据的数据集。网络流量捕获文件的来源是CICIDS-2018，核心方法是流量分割方法，将pcap文件按照规则分割为一个个样本。按照计算机网络的基本概念，本文将异常入侵检测判断的单个样本定为会话（session），进行流量分割。最后构建了一个80万个样本的会话样本集，并对其进行了可视化分析，将各种异常会话样本以黑白图像的形式可视化，最后得出这些会话的二维图像具有明显的图形特征。在第三小节中则对数据集中数据包个数分布，会话字节分布，会话持续时间分布做出了深入分析，论证了本文提出的流量分割方法的合理性，展示了数据集的几个关键特征的统计规律。

1. 基于深度学习的端到端的异常入侵检测算法

在第3章中本文详细介绍了构建一个会话样本字节数据的网络流量数据集全过程，上述数据集是本章算法的重要数据来源。在深度学习中，数据的形式往往直接决定了使用何种深度学习模型。图像使用卷积神经网络，文字使用循环神经网络，图数据使用图神经网络。网络流量数据可以被分割成一系列数据包组成的会话样本，每个会话样本包含的一系列数据包是有先后顺序的序列关系，对于序列数据循环神经网络非常适合处理；将全部数据包融合而成的二维图像可以使用卷积神经网络处理，得到样本的整体特征。由于网络数据是遵从相同协议下具有固定格式的数据，在对会话图像的可视化分析中看到，会话的可视化图像具有明显的图形规律。

本章将详细介绍如何构建一个包含两种不同神经网络的深度学习算法，用于异常入侵检测。在本章算法中，一端输入是会话样本数据，它是直接由流量分割得到的，经过简单处理，一端是异常入侵检测任务模型，判断会话样本是正常会话还是异常会话，这种一端是数据，一端是任务的结构就是端到端。

4.1 算法

在4.1节中，将会详细介绍本文提出的异常入侵检测算法的深度学习模型，并介绍优化算法，超参数调整。

4.1.1 模型整体结构

本文算法的模型的简化示意图如图4.1所示，包括四个模块，卷积模块，循环模块，注意力模块，和最后的分类器模块。卷积模块用于处理融合后的会话图像，循环模块处理会话包含的一系列数据包，注意力模块根据循环模块的输出给卷积模块输出的特征图不同区域生成不同权重，最后连接加权的卷积特征图与循环模块输出送入分类器中进行分类。



图4.1 提取会话样本图形特征的卷积网络结构图

Fig. 4.1 the simple overview of my model’s architectures

4.1.2 卷积模块

在卷积网络结构中需要考虑如下的因素，其一是卷积核的大小选择，二是模型的深度，三是否改变卷积的结构增加跳跃连接等。如今，计算机视觉领域在卷积神经网络上的研究非常深入，在各种特定应用上都有着高性能的模型，本文基于卷积网络的基础结构，用卷积层，池化层完成对会话样本融合而成的二维图像的特征提取，如图4.1为适用卷积神经网络进行会话图形特征提取的网络结构图。

 （4.1）

式中，I为会话图像张量。为（512，4，4）的三维张量，为卷积模块的最终输出。



图4.2 卷积模块网络结构图

Fig. 4.2 the architectures of Convolutional module of model

如图4.2所示，模型的输入是三维张量I（1，32，32），张量的三个维度代表的含义是（通道数c，高度h，宽度w），首先经过一个3\*3卷积层，输出张量为（32，30，30），再经过一个2\*2的最大池化，输出张量为（32，15，15）。再将输出张量送入3\*3卷积，2\*2池化，2\*2卷积，输出一个（128，4，4）的三维张量。最后将这个（128，4，4）张量送入2\*2卷积层，输出（512，4，4）的张量。

如表4.1展示了算法的卷积网络结构,以及各层的具体参数。

表4.1 算法的卷积网络结构层次

Table4.1 the Table of convulution architectures

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Layers** | **Output size** | **channel** | **kernel** | **stride** |
| Convolution | 30\*30 | 32 | 3\*3 conv, | 1 |
| Maxpool | 15\*15 | 32 | 2\*2 max pool | 2 |
| Convolution | 13\*13 | 64 | 3\*3 conv | 1 |
| Maxpool2 | 6\*6 | 64 | 2\*2 conv | 2 |
| Convolution3 | 5\*5 | 128 | 2\*2 conv | 1 |
| Convolution4 | 4\*4 | 512 | 2\*2 conv | 1 |
| Dropout | 4\*4 | 512 |  |  |

4.1.3 循环模块

在网络流量会话样本中，包含有一系列的数据包，每个数据包最多包含320个有效字节，这些具有先后顺序关系的数据包可以被视为一个整体的序列数据。有一个较为合理的解释是：每一个数据包被表示维一个320维长度的向量，类似于一个单词在320维空间的嵌入向量，一系列数据包就表示两个通信主机之剑的对话，为一段文字。因此适用循环神经网络来理解这段话，并且识别记忆这段话的关键信息，成为判断会话是正常会话或是异常会话的关键点。

**P**={p1,..,pn},这里pi为每个数据包向量，**P**是所有的数据包向量集合。

 （4.2）



图4.3 提取序列特征的循环模块

Fig. 4.3 the Recurrent module for extracting sequence features

图4.3展示了序列特征提取网络。这是一个循环神经网络，使用门控循环单元（GRU）作为基础单元。从图可以看到，我们的输入一系列数据包，数据包表示为320维的向量，每次数据包输入，都会输出一个隐藏状态hi，取最后一次输出的隐藏状态hn作为对该序列信息的特征向量，该向量为512维。

在前述第3章构造的会话样本集中，一个会话最多取32个数据包，因此在序列提取网络中，网络最多循环计算32次，输出32个隐藏状态，分别对应第i个数据包时网络的信息积累状态。

4.1.4 注意力模块

如果将模型的二维会话输入视为图片，将一系列数据包视为一句话，本文模型的输入结构类似于一个图片问答系统[33]。想象一下，对于会话图像，我们的根据循环模块的输出来“看”图像的某个区域，对于不同区域显然应该有不同的重要程度，以此来生成注意力权重，将这个权重与卷积模块输出的特征图相乘就得到注意力图，将这个注意力图与循环模块的输出再次送入注意力层，就可以在第一次注意力图的基础上生成第二次注意力图，显然在不断的“注意”下会排除无关区域，聚焦于更相关的区域，堆叠的次数取决于问题的复杂程度。而这就是堆叠注意力网络SAN（stacked attention networks）[34]的思想。

SAN的公式如下，

 （4.3）

 （4.4）

 （4.5）

 （4.6）

式中：

代表逐元素相加，

**vI**是卷积模块输出,

**vP**是循环模块输出,

WI,A, WP,A, Ws是三个不同的权重,

pI是softmax输出的概率值向量。

d是会话融合图像的通道数，也是循环模块最后输出的向量的维数，m是会话图像I输出特征图的尺寸，例如卷积模块输出的特征图，则应该转换为，即m=w\*h=16。

经过以上4.3式和4.4式的计算我们得到，是一个m维的向量，对应的就是卷积模块特征图的对应区域，用这个向量中的概率和特征图每个对应区域的向量相乘就得到一个加了权重的注意力图，将其与相加得到，将u视作新的送入式4.3，进行新的一轮运算就能得到新的注意力图和ut,t为运算迭代的次数，也就是注意力网络堆叠的层数。

在实践中，该注意力网络的实现方式和理论上略有不同，注意力图根据不同的初始化卷积核来进行生成(意味着不需要迭代，一次生成，相当于看两次，而不是在第一次看的区域上缩减范围，但效果相同)，网络结构如图4.4。

卷积模块输入会话融合图像I，输出特征图**vI**;循环模块输入一系列数据包（p1，…,pn）输出特征向量vP，将（vI，vP）送入注意力模块。输出一个向量u。



图4.4 注意力模块

Fig. 4.4 the Attention module of model

式（4.3）中，WI,AvI是先对vI(512,4,4)进行变换，如下图4.4中4.3所指上半所示，使用的是有着3\*3卷积核的卷积变换，最后变换完为WI,AvI，WP,AvP +bA表示对vp进行变换，如下图对应4.3所指下半部分，变换完后WP,AvP +bA。代表要将变换后两个量逐元素相加，vI变换后是一个张量，但是vp变换后是一个向量，因此需要对变换后vp进行扩展，具体就是从到，例如vp变换后向量第一个元素是1，那么就会扩展为[[1, 1, 1, 1],[1, 1, 1, 1],[1, 1, 1, 1],[1, 1, 1, 1]],依次类推到所有元素，最后就能与变换后vp逐元素相加得到hA。

式（4.4）中，如下图对应4.4所指先使用1\*1的卷积，融合通道信息，输入hA变换后为，先展平为，然后使用softmax函数，得到pI。其中g是注意力层堆叠层数。本模型取g=2。

式（4.5），如图对应4.5所指，使用权重pI乘（矩阵乘法）以图像特征vI的每个区域vi,得到注意力特征图，在编程实现中，具体步骤如下：首先将vI的空间维度展平，变为vI，对vI应用注意权重pI，得到注意力attention=，然后将attention第三维度加和得到,最后展平得到一个向量a=，该向量a就是注意力模块的输出。

式4.6，将a与v在第一个维度连接，得到u。

可以看到注意力网络的堆叠次数通过第二次卷积最后留下的通道数g来控制。g是英文glimpse的缩写，即“瞥见，看一眼“的意思，因此这种实现的堆叠注意力网络更像是看几次，集中在特征图的不同区域，而不是逐渐聚焦于某个区域。

4.1.5 分类模块和总结

分类模块为两个全连接层+softmax层。

 （4.7）

如图4.5所示，将注意力网络输出u送入两个全连接层，将其映射到输出空间，如果是二分类就是二维，如果是多分类就是类别数，本文算法有15个类别，因此会映射到15维的空间，输出一个15维的向量，使用softmax函数得到每个类别的概率值，取最高概率所在类别得到模型预测。



图4.5 分类模块

Fig. 4.5 the Classifier module of model

1. 总模型

在我们的会话样本中，样本被存储为n\*320的张量文件，取前1024个有效字节（非0填充）融合为会话图像送入卷积网络，取单独的数据包作为1\*320的向量送入循环神经网络，然后将这两个输出送入注意力模块。

我们取注意力层看的次数为2，生成了两个注意力图，将这两个注意力图和循环模块输出的特征向量拼接concatenate，得到融合特征向量u，将u送入一两个全连接层组成的分类模块中，最后输出一个1\*categorys（categoryes取决于模型需要分类的类别），最后使用softmax函数输出模型对某个类别预测概率，取最高概率所在的类别，就是模型对会话样本的预测结果。

如图4.3所示，展示了本章要阐述的端到端异常入侵检测算法模型的整体网络结构。



图4.6 端到端的异常入侵检测算法模型的网络结构

Fig. 4.6 the End-to-end anomaly intrusion detection model network architectures

如图4.6所示，该算法的在注意力模块前两个特征提取网络是平行的，互不干扰。最后将两个网络的输出融合作为会话样本的总特征向量，经过全连接变换到输出维度，经过softmax输出预测概率。

算法的重点是对会话样本包含的信息进行提取，从较低级的信息中提取高级的语义特征，从而将检测会话样本。可以看到，算法使用的模型并不复杂，但是实验的参数选取需要谨慎考虑。

4.1.6 损失函数和优化算法

1. 交叉熵损失函数

在4.1.1中的异常入侵检测算法模型，最后会输出一个类别概率向量。假设模型执行二分类任务，那么输出的向量可能是（0.87，0.13）。如果我们规定0代表正常，1代表异常，那么模型的最终预测是取概率最高所在的类别，也就是下标，在该假设下，模型输出该会话样本为正常的概率为0.87，异常的概率为0.13，因此认为模型预测该会话样本为正常样本。

但是，我们怎么判断模型这次预测是好的，还是坏的，这就需要一个准则来告诉我们模型的好坏，这就是损失函数。在执行多分类任务的深度学习模型中，最常用的准则是最大似然估计[35]。

假设有m个样本的数据集*X*={x(1), x(2), …, x(m)},数据集上的经验分布为pdata。我们定义了上述的深度学习模型,模型的分布为pmodel。

而最大似然估计就是最小化训练集上的经验分布和模型分布之间的差距。模型分布,对于模型参数的最大似然估计为式（4.8）：

 （4.8）

由于概率乘积不便于计算且会出现数值下溢（即数值趋于0），因此我们将其取对数，取对数不会改变式子的求解，是等价优化，如式（4.9）：

 （4.9）

 同样，缩放上述式子，也不会改变argmax，为此为了将模型分布与数据集的经验分布相关联，可以除以数据集个数m，如式（4.10）：

 （4.10）

对变量*X*求的均值，就是求在X~Pdata情况下的期望（根据切比雪夫大数定律，当样本趋于无穷时，算术平均值依概率收敛于数学期望），因此(4.3)式等价于(4.4)：

 （4.11）

同时最大化就是最小化-，因此最终我们得到如下函数：

 （4.12）

上述函数就是交叉熵（cross entropy）损失函数。

此外KL散度表示同一随机变量两个分布之间的差异程度，定义如下：

 （4.13）

可以看到式（4.6）的右半部分的后一项就是交叉熵。因此最大似然函数估计的理解就是最小化模型分布和经验分布之间的交叉熵。

因此，本文使用的损失函数就是交叉熵损失函数，在输入样本后，计算损失函数，使用优化算法最小化该函数，就是模型的训练过程。

假设在模型执行二分类的情况下，输入一个样本，模型输出为pmodel（0.8，0.2），模型为正常样本，因此实际概率分布为pdata（1，0），则模型在该样本下的交叉熵计算公式4.14如下：

 （4.14）

因此本次的模型损失为0.09691，损失相对较小，模型当前性能较好。

1. 优化算法

在上一节中，损失函数代表模型的目标函数，模型训练的目的就是最小化损失函数来接近真实分布，关于如何优化目标函数，深度学习广泛使用基于梯度下降的学习算法。在微积分中，函数的梯度告诉了我们函数的变化趋势（假设有个函数y=f(x),当梯度为正，较少x可以减小y；为负，增加x可以减少y），因此，我们只需要逆梯度的趋势减少参数值，就可以达到减少失函数值的目的。如果求出损失函数关于各个参数的梯度，然后根据梯度我们只需要用一个学习率乘以梯度，然后计算，那么根据微积分的知识，损失函数就会减少，逐渐的接近最小值。

上述是随机梯度下降算法的基本原理。在计算梯度时，一般采用随机小批量来计算梯度。小批量是从训练集中均匀抽出的小批样本B={ x(1), x(2), …, x(b)}（相对于数据集很小，一般不过百），用着几个样本计算的梯度来近似估计整个训练集在当前模型状态下的总梯度，即数学期望。

计算公式为式4.15：

 （4.15）

式子中，L为损失函数。b为小批量样本数目。理论上来说，每次都计算整个训练集上每个样本的梯度来优化模型是最理想的，然而计算代价实在太大。使用小批量估计梯度的理论解释是：由于m个样本均值的标准差为，可以看到根据分布，增加样本数，标准差的减少曲线低于线性，因此使用小批量估计的方法快速计算梯度，模型收敛的速度会更快的收敛；此外我们的训练集往往不是每个样本都非常独特，缺少它据集表达的分布就离真实分布更远，因此可以近似的认为大部分样本对梯度的贡献是相似的，差异不大，因此从小批量样本估计梯度具有一定合理性。

事实上，上式更新参数的方法被称为随机梯度下降，是最基本的优化算法，但是模型和学习率之间的关系非常固定，就是每次都沿着梯度下降固定的值，这对于模型优化是不利的，有时候在不同的参数值情况下可能需要不同的学习率，为此研究者提出了学习率自适应的优化算法，一种算法是Adam，假设目标函数参数为，一阶矩变量是，二阶矩变量r=0，根据式4.15计算小批量样本梯度g，然后更新公式如下：

 （4.16）

 （4.17）

 （4.18）

 （4.19）

 （4.20）

 （4.21）

式中，和是s，r两个矩估计的衰减速率，默认分别为0.9和0.999。是为了防止出现除0错误，用于数值稳定的常数，默认为10-8。为逐元素应用。和是经过修正的的一阶和二阶矩估计。

从式4.16和式4.17看出一阶矩s和二阶矩r都和历史值有关，而阶数表明了与梯度的关系。在不断累积历史梯度的情况下，根据当前梯度值自适应的改变参数更新幅度，研究表明Adam优化算法具有相当的鲁棒性。

4.2 实验设计

4.2.1 实验目的和步骤

实验目的：在同一个数据源的基础上（CICIDS2018）,对比两种不同入侵检测方法的性能。两种方法分别是基于会话样本的使用原始网络流量数据的入侵检测方法，通过深度学习自动学习特征；一种是原始网络流量数据提取专家定义的人工特征，基于人工特征集的入侵检测方法。

实验步骤：

1.实验环境准备，包括实验所用到所有程序和工具。

2.下载基于ciciids2018的会话样本集和人工特征集，编写数据加载和预处理程序（对于人工特征集有两个预处理，类别特征变为one-hot向量，对数值特征的归一化处理）。

3.按照4.1所示模型编写代码，代码框架为pytorch，获得深度学习模型

4.运用scikitlecrn的库，设置svm，随机森林等算法的具体参数

5.输入数据，训练深度学习模型和其他算法模型，得到结果。通常结果就是模型对每个样本的预测类别，一个0~14的整数，以矩阵形式存储算法结果。

6.按照下文4.3.1所介绍的评价指标，计算模型的性能。

7.通过评价指标计算结果对模型性能进行分析。

4.2.1 实验环境和学习率设置

本次实验在个人pc上进行。系统为win11-21H2,操作系统版本为22000.434。cpu处理器为：i5-10400 CPU @ 2.90GHz。显卡gpu：GTX1060 。内存大小：16GB。

实验代码均使用python语言编写，版本为：python-3.8.10。下表4.2展示实验使用的程序包和版本：

表4.2 实验使用的程序包

Table 4.2 the Packages used in experiments

|  |  |  |  |
| --- | --- | --- | --- |
| **程序包** | **版本** | **功能** | **是否开源** |
| Numpy | 1.20.3 | 数值计算 | 是 |
| Pandas | 1.3.4 | 数据分析处理工具 | 是 |
| Scikit-learn | 1.0.1 | 开源机器学习，优秀的算法实现 | 是 |
| Pytorch | 1.9.0 | 开源机器学习框架，深度学习 | 是 |
| Matplotlib | 3.5.0 | 数据可视化，绘图框架 | 是 |

在本次实验中，使用pytorch框架构建我们的深度学习模型，损失函数为交叉熵损失函数，优化算法为Adam优化算法，重要超参数是学习率，关于在初始时设置为0.001。

而在训练过程中，采用余弦曲线衰减的方式，减小学习率。如图4.7所示：

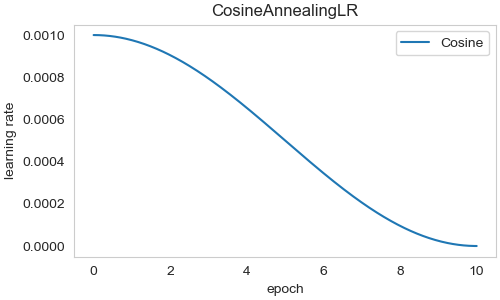


图4.7 学习率的余弦衰减曲线

Fig. 4.7 the figure of CosineAnnealingLR

计算公式4.22如下：

 (4.22)

式中，为当前epoch时的学习率， 是初始学习率也是最大学习率，是最小学习率，Tcur是当前epoch，Tmax是最大epoch（周期）。如图4.4所示为初始学习率为0.001，Tmax=10，时学习率的变化曲线。可以看到该曲线衰减的特点是前期衰减少，中期衰减多。

4.2.3 数据和参数设置

设置两组实验，一组是使用第三流量分割得到的数据集使用4.1节介绍的模型进行训练，验证模型效果，为端到端的实验。

一组是采用cic-ids208人工特征集使用传统机器学习算法的入侵检测实验。为非端到端的实验。

在对比实验中，对比的点有：1，端到端的深度学习方式，直接使用网络流量数据作为模型输入，进行异常入侵检测。而使用人工特征集进行入侵检测前需要对收集到的网络流量数据进行统计分析，提取人工特征，不具有实时性，费时费力。2，实时性，本文提出的入侵检测算法针对网络会话数据，提取最多32个数据包即可送入网络进行检测，而且只针对单个会话流量。人工特征数据集的方法需要收集到整个会话的流量，甚至是一天的流量进行统计分析，才能得出一个会话的特征，不具备实时检测的效果。

两组实验基于的数据均来源于cic-ids2018。

1. 使用网络流量数据集的端到端的基于深度学习的实验

数据集：第三章的成果，样本类别和分布如下表4.3

表4.3 对比实验1数据集分布表

Table 4.3 the table of Comparative experiment 1 data set

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编号** | **类别** | **样本数** | **编号** | **类别** | **样本数** |
| 0 | Benign | 388898 | 8 | DDoS- LOIC-UDP | 110 |
| 1 | FTP-BruteForce | 29599 | 9 | DDOS-HOIC | 99727 |
| 2 | SSH-Bruteforce | 36510 | 10 | Brute Force -Web | 137 |
| 3 | DoS-GoldenEye | 25323 | 11 | Brute orce -XSS | 43 |
| 4 | DoS-Slowloris | 6979 | 12 | SQL Injection | 17 |
| 5 | DoS-Hulk | 14116 | 13 | Infiltration | 88144 |
| 6 | DoS-SlowHTTPTest | 14116 | 14 | Bot | 46571 |
| 7 | DDoS LOIC-HTTP | 49703 | All | 总计 | 80,0000 |

总计15个类别，80万个样本。使用4.1.1图4.1所示模型，在此数据集上进行实验,调试模型参数，分别对Dropout，循环周期epoch等不同超参进行三次对比实验。

具体实验结果见4.3.2。

1. 使用人工特征集的使用传统机器学习的实验

数据集：官方使用flowmeter（一种流量处理工具）提取的有80维人工特征数据集。选取一定数量样本，类别分布如表4.4。

表4.4 对比实验2数据集分布表

Table 4.4 the table of Comparative experiment 2 data set

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编号** | **类别** | **样本数** | **编号** | **类别** | **样本数** |
| 0 | Benign | 449638 | 8 | DDoS- LOIC-UDP | 1730 |
| 1 | FTP-BruteForce | 25000 | 9 | DDOS-HOIC | 48270 |
| 2 | SSH-Bruteforce | 25000 | 10 | Brute Force -Web | 249 |
| 3 | DoS-GoldenEye | 40000 | 11 | Brute orce -XSS | 79 |
| 4 | DoS-Slowloris | 10000 | 12 | SQL Injection | 34 |
| 5 | DoS-Hulk | 25000 | 13 | Infiltration | 50000 |
| 6 | DoS-SlowHTTPTest | 25000 | 14 | Bot | 50000 |
| 7 | DDoS LOIC-HTTP | 50000 | All | 总计 | 80,0000 |

总计15个类别，共80万个样本。使用传统机器学习方法进行分类实验，使用的传统机器学习算法如下表4.5：

表4.5 对比实验使用的传统机器学习算法及其参数

Table4.5 the table of Comparative experiment of used machine learning algorithms and their parameters

|  |  |
| --- | --- |
| **算法** | **参数** |
| 决策树 | 基于CART |
| 随机森林 | 最大100棵树 |
| 贝叶斯 | 基于高斯核 |
| Adboost | 100个弱分类器 |
| 支持向量机 | 核函数：rbf |
| 最近邻kNN | 邻居数为5 |
| 多层感知机 | 隐藏单元256，迭代300次 |

在上表中一些算法的其余参数，均为默认情况，并且可以看到在表最后一列，为一个多层感知机（一种简单的神经网络模型），来看简单的神经网络模型在人工数据集中的表现。具体实验结果见4.3.3。

数据集的划分：两个对比实验均按照训练集和测试集7：3的比例划分，因此训练集56万个，测试集24万个。

4.3 实验结果与分析

4.3.1 评价指标

混淆矩阵（confusion matrix），它可以用来评价模型、算法在分类任务上的性能表现。如表4.6展示了二分类情况下的混淆矩阵。

表4.6 混淆矩阵

Table4.6 the Confusion Matrix

|  |  |  |
| --- | --- | --- |
|  | 预测为真 | 预测为假 |
| 实际为真 | True Positive (TP) | False Negative (FN) |
| 实际为假 | False Positive (FP) | True Negative (TN) |

准确率（accuarcy）: 准确率(accuarcy)：模型预测正确的数目占所有样本的比例，如公式4.23所示。在各个类别平衡时，该指标能够表明模型性能，但是在极端类别情况，如正类10个，负类990个，模型只需要输出全负类，就可以达到99%的准确率，然而对正类的预测能力为0。

 （4.23）

精确率（precision）：模型预测为类别C的样本中实际类别为C的样本所占比例，如公式4.24

 （4.24）

召回率（recall）:实际类别为C的样本中模型预测为C的样本所占比例，如公式4.25。在上面阐述的类别不平衡分类中，就可以计算出正类的召回率为0/10=0,由此可以判断模型的预测能力完全无效

 （4.25）

F1值（f1-score）：精度 (p) 和召回率 (r) 的调和平均值，如公式4.26，它综合考虑了precision,recall的值，F1值越高，模型性能越好。

 （4.26）

对于多分类的情况，如表4.7所示。展示了三类情况下的混淆矩阵。

表4.7 多分类混淆矩阵

Table4.7 the Multliclass Confusion Matrix

|  |  |  |  |
| --- | --- | --- | --- |
|  | **A** | **B** | **C** |
| **A** | 2 | 1 | 3 |
| **B** | 3 | 2 | 4 |
| **C** | 1 | 3 | 5 |

对于准确率，只能计算所有类别的准确率，计算公式acc=对角线上元素相加/表格所有元素之和。

类别S的精确率precission=（actual=pred=s）/预测为S所在列元素之和。

类别S的召回率recall=（actual=pred=s）/实际为S所在行元素之和。

F1-score的计算公式不变。因此混淆矩阵可以很简单的扩展到多分类的情况。

4.3.2 基于会话样本的入侵检测实验

数据集为第三章构造的数据集，为80万个含有原始流量字节数据的会话样本，使用的模型为图4.1.1所示。本实验组总计三次实验，分别对应模型不同的超参数情况。小标题中黑色下划线为对照变量。

1. 对照实验1，Dropout=0.5，训练集迭代epoch=10，batch=32

Dropout是一种正则化模型的方法，可以看成一种计算简单而强大的Baaging集成近似，能够训练和评估指数级数量的神经网络。具体操作上就是对于输入单元，以概率p进行采样。例如一个1024维的输入上应用dropout，设置采样概率为0.5，那么表明这1024个输入单元中每一个单元只有1/2的概率会被包含，参与到下一层的计算中，如果采样次数足够，应该是有512个单元被采样激活，进入下一层神经网络计算。

通过以固定概率丢弃一些输入神经单元不参与计算，那么在每一次训练学习时，整个模型其实只训练了一个基础网络的子网络，因此使用Dropout训练就相当于集成了一个基础网络上的子网络模型集。

在此次实验中，我们对图4.1所示模型的所有模块应用Dropout机制。卷积模块最后输出应用一层Dropout，循环模块初始输入应用一层Dropout，注意力模块所有变换之后均有一层Dropout，分类模块在输入前后第一个全连接层后也均有一层Dropout，设置Dropout rate=0.5（在4.1.1中可以详细看到位置）。

实验结果如下表4.8.

表4.8 dropout=0.5，epoch=10模型实验结果表

Table4.8 the table of dropout=0.5, epoch=10 model experiment result

|  |  |
| --- | --- |
| **指标** | **结果(%)** |
| Accuracy | 97.30 |
| Recall | [94.51, 99.99, 99.99, 100.0, 99.65, 100.0, 100.0, 99.99, 89.55, 99.99, 86.67, 81.48, 20.0, 99.89, 99.84] |
| Precision | [99.94, 100.0, 99.99, 99.98, 98.37, 100.0, 99.99, 99.99, 92.31, 100.0, 87.5, 91.67, 40.0, 99.70, 68.87] |
| F1-score | [97.15, 99.99, 99.99, 99.98, 99.01, 100.0, 99.99, 99.99, 90.91, 99.99, 87.08, 86.27, 26.67, 99.80, 81.51] |

在表4.8中，除准确率acc外，对应结果一栏中[]内每一项对应一个类别的指标值，类别序号为下标，对应的真实类别名称在表4.3可以看到。

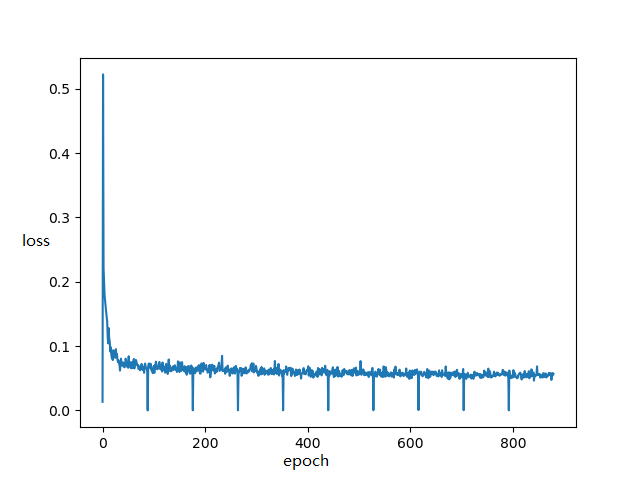


图4.8 实验1损失函数曲线图

Fig. 4.8 the figure of loss changes of compare experiment 1

图4.9深度模型实验1结果柱状图

Fig. 4.9 the histogram of Depth model control experiment 1 results

图4.8展示了损失函数值的变化情况，在该图中，横坐标为训练迭代的次数epoch。本次模型训练中批次为32个样本，且每训练200次记录一次损失函数值，训练集大小为560000，而560000/（32\*200）=87.5，因此横坐标每过87.5代表训练集已经遍历完一次，过了一个epoch,图中记录了训练集迭代10次损失函数的变化值，可以看到，迭代两次epoch=2（175）时损失函数就已经趋于收敛，之后下降的幅度非常小。

从图可以看出对于类别8：DDoS- LOIC-UDP，10：Brute Force -Web，11：Brute orce -XSS，12：SQL Injection，14：Bot的分类检测上，效果差于其他攻击类别，其中8，10，11，12类别样本稀少时，是造成模型对该类攻击识别较低的原因。对于攻击12:SQL injection即sql注入攻击检测的三个指标均很低，可能是由于样本最少的缘故（只有17个）。

表4.9中表明了模型的总体准确率为97.30%，但是从损失函数曲线，和模型指标表现，分析认为模型参数可以进一步调整，以提高性能。

1. 对照实验2，Dropout=0.2，epoch=10，batch=32

在对照实验1中，我们设置dropout=0.5，并且在所有模块都应用了Dropout机制。Dropout本质上减少了模型容量，并且需要更多的迭代次数才能收敛。而本文算法的模型并不复杂，层数也不够深，即模型容量不是很大，因此为了更快收敛和减小Dropout机制对模型容量的影响，考虑增加Dropout的采样概率，设置Dropout rate=0.2，表明模型将以20%的概率丢弃输入单元，也就是以80%的概率采样输入单元。

我们重新审视一下我们构建的模型，看一下我们模型的输入是什么。注意到我们模型卷积模块的输入是1\*32\*32的单通道黑白图像，是会话样本前1024个字节的融合图像。图像的像素是32\*32，分辨率较小。且该卷积模块也非常简单，在简单模型上应用dropout机制集成的子网络模型性能可能非常低，反而降低了总体的效果，因此在卷积模块，我们设置Dropout=0.0，即保持原始输入。

对于循环模块，我们的输入是一系列数据包，每个数据包的维数是320。对于自然语言处理使用的循环神经网络，我们输入的是一句话在词库中的索引，然后经过嵌入层映射为嵌入向量，嵌入向量是一个单词在高维空间的向量表示。对于该特征向量运用Dropout有助于学习到更加优秀的嵌入式表示。同理这也是基于更大的模型容量，学到更好的嵌入式表示。但是，本文模型输入是数据包的原始字节数据，并不是索引经过嵌入层后的嵌入向量。对于数据包来说，例如头部字节的某一位如果改变，意义完全不一样，因此在循环模块也设置Dropout=0.0，即对循环模块输入保持原始输入。

对于注意力模块、分类模块设置Dropout=0.2。



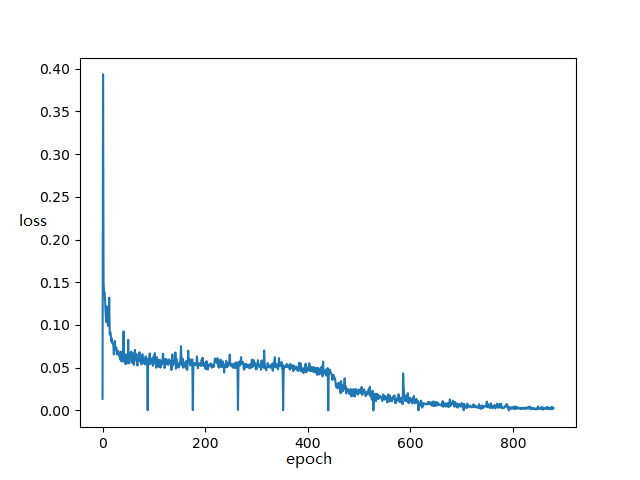


图4.10 实验2损失函数曲线图

Fig.4.10 the figure of loss changes of compare experiment 2

**** 如图4.10所示，横坐标为周期epoch，纵坐标为损失函数值loss。为设置Dropout=0.2时实验的损失函数曲线图，从图中可以看到明显的二个平台期（趋于直线的部分），一个在epoch=2的时，一个大概在epoch=5（横坐标=437）时。因此和实验1相比，本次模型在迭代训练集5次后，迎来了一次模型损失函数的下滑点，因此可以预测模型的性能指标比实验1好。

表4.9展示了Dropout=0.2时的实验2的结果。

表4.9 dropout=0.2，epoch=10模型实验结果表

Table4.9 the table of dropout=0.2, epoch=10 model experiment result

|  |  |
| --- | --- |
| **指标** | **结果(%)** |
| Accuracy | 99.93 |
| Recall | [99.88, 100.0, 100.0, 100.0, 99.71, 100.0, 100.0, 100.0, 100.0, 99.99, 99.05, 92.59, 90.0, 99.98, 99.79] |
| Precision | [99.97, 100.0, 100.0, 100.0, 98.71, 100.0, 100.0, 99.99, 94.37, 100.0, 97.20, 96.15, 100.0, 99.81, 99.61] |
| F1-score | [99.93, 100.0, 100.0, 100.0, 99.21, 100.0, 100.0, 99.99, 97.10, 99.99, 98.11, 94.33, 94.74, 99.89, 99.69] |

表4.9中除准确率外，其余指标[]中每一项都对应某一类攻击在该指标下的值，对应关系为：{'Benign': 0, 'FTP-BruteForce': 1, 'SSH-Bruteforce': 2, 'DoS attacks-GoldenEye': 3, 'DoS attacks-Slowloris': 4, 'DoS attacks-Hulk': 5, 'DoS attacks-SlowHTTPTest': 6, 'DDoS attacks-LOIC-HTTP': 7, 'DDOS attack-LOIC-UDP ': 8, ' DDOS attack-HOIC ': 9, 'Brute Force -Web': 10, 'Brute Force -XSS': 11, 'SQL Injection': 12, 'Infilteration': 13, 'Bot': 14}。

下图4.11展示了这些指标的柱状可视化图。

图4.11深度模型对照实验2结果柱状图

Fig. 4.11 the histogram of Depth model control experiment 1 results

从图4.11可以看出，模型分类结果中对8：DDoS- LOIC-UDP，10：Brute Force -Web，11：Brute orce -XSS，12：SQL Injection，和其他类的检测效果明显较低，这也是之前模型存在的问题，是样本数较少的缘故。但是从结果上来看，相比上一次在这几个类别上较差的结果，最低对12：SQL Injection只有26%左右的召回率和40%的精确率，这次实验中对SQL Injection的检测在三个指标上均达到了90%以上的高数值，证明模型性能获得了一定提升。

从准确率上来看，本次实验模型的总体准确率从97.30%至99.93%,也有一定提升。

1. 对照实验3，Dropout=0.2，epoch=20，batch=32，学习率衰减T=20

实验结果如下表4.8，模型准确率为99.96%。

表4.10 dropout=0.2，epoch=20模型实验结果表

Table4.10 the table of dropout=0.2, epoch=20 model experiment result

|  |  |
| --- | --- |
| **指标** | **结果(%)** |
| Accuracy | 99.959 |
| Recall | [99.94, 100.00, 100.00, 100.00, 99.94, 100.00,100.00, 100.00, 98.51, 100.00, 98.10, 92.59, 100.00, 99.95, 99.89] |
| Precision | [99.97, 100.00, 100.00, 100.00, 98.77, 100.00, 100.00, 100.00, 94.29, 100.00, 98.10, 96.15, 100.00, 99.92, 99.88] |
| F1-score | [99.96, 100.00, 100.00, 100.00, 99.35, 100.00, 100.00, 100.00, 96.35, 100.00 ,98.10 ,94.34, 100.00, 99.94, 99.88] |

如下图4.12为该次实验的损失函数曲线变化。本次实验对训练集迭代了20次，可以看到在迭代15（横坐标=1300）次后，目标函数逐渐收敛，函数曲线趋于直线。

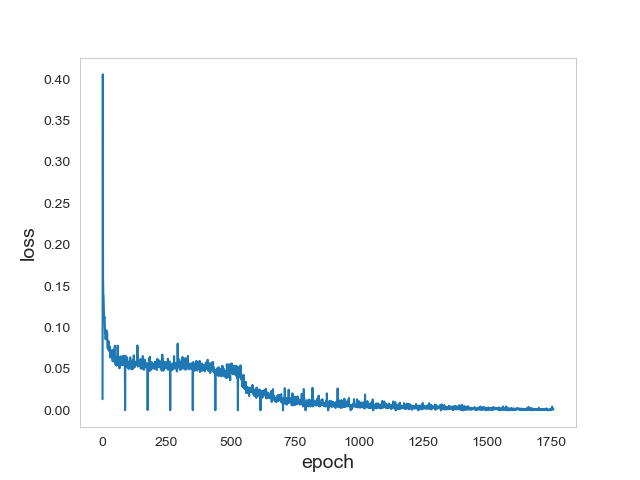


图4.12 实验2损失函数曲线图

Fig. 4.12 the figure of loss changes of compare experiment 2

下图4.8为模型对15个类的评价指标柱状图。可以看到，模型仍然对8：DDoS- LOIC-UDP，10：Brute Force -Web，11：Brute orce -XSS，12：SQL Injection这四种攻击的检测效果较差，但仍然达到了平均值92%以上得相对好成绩，比前两次对比实验好。

图4.13深度模型对照实验3结果柱状图

Fig. 4.13 the histogram of Depth model control experiment 3 results

如图4.13，模型得主要优化均在这几个样本较少的类上。对于样本较少的类需要更多的样本来学习才能更加准确的检测。但是模型一直对是否是正常会话样本保持了很高的准确率（F1-score:99.96%），这说明模型完全可以根据判断该样本是否是正常样本来高精度的检测到异常样本（非正常），而无需确切异常样本的数据，即异常检测。

4.3.3 基于人工特征的入侵检测实验

实验使用官方提供的人工特征数据集。选取80万样本，共计15个类别，数据集具体分布见表4.2。使用的传统机器学习方法见表4.3。

实验结果如下表4.11：

表4.11人工特征数据集的传统机器学习实验结果表

Table4.11 the experiment result table of traditional machine learning experiments on artificial feature datasets

|  |  |
| --- | --- |
| **决策树（CART）** | |
| **Acc** | 0.9118 |
| **recall** | [0.9405, 0.8941, 0.9997, 0.9999, 0.9902, 0.9997, 0.5199, 0.9989, 0.9732, 1.0000, 0.8806, 0.8571, 0.7000, 0.4251, 0.9983] |
| **precision** | [0.9357, 0.6543, 0.9995, 0.9995, 0.9979, 0.9996, 0.8277,0.9986, 0.9713, 0.9996, 0.9672, 0.9473, 0.6363，0.4454, 0.9971] |
| **F1-score** | [0.9381, 0.7556, 0.9996, 0.9997, 0.9941, 0.9996, 0.6387, 0.9987, 0.9722, 0.9998, 0.9218, 0.90, 0.6667,0.4350, 0.9977] |
| **随机森林** | |
| **Acc** | 0.919475 |
| **recall** | [0.9844, 0.6152, 1.0, 0.9995, 0.9086, 1.0, 0.8245, 0.9428, 0.9595, 1.0, 0.0093, 1.0,0.0, 0.1334, 0.7700] |
| **precision** | [0.9357, 0.6543, 0.9995, 0.9995, 0.9979, 0.9996, 0.8277,0.9986, 0.9713, 0.9996, 0.9672, 0.9473, 0.6363，0.4454, 0.9971] |
| **F1-score** | [0.4558, 0.7349 ,0.9994, 0.9985 ,0.9456, 0.9974, 0.5559, 0.9692, 0.9793, 0.9997, 0.0184, 0.8333, 0.0, 0.2335, 0.6409] |
| **贝叶斯(高斯内核)** | |
| **Acc** | 0.1823625 |
| **recall** | [0.0249, 0.3398, 0.5059, 0.1500, 0.6827, 0.0017,0.8819, 0.4856, 0.9904, 0.9999, 0.0597, 0.4761,0.6, 0.0771, 0.0] |
| **precision** | [0.9547, 0.6085, 0.3812, 0.3949,0.2899, 0.0032, 0.3989, 0.4311,0.9574, 0.1226, 0.0167, 0.0001,0.0, 0.0, 0.0] |
| **F1-score** | [0.0486, 0.4361, 0.4347, 0.2174,0.4069, 0.0023, 0.5493, 0.4567,0.9736, 0.2184, 0.0261, 0.0018,0.0004, 0.1004, 0.00] |
| **AdaBoost** | |
| **Acc** | 0.5609416666666667 |
| **recall** | [0.8845, 0.0, 0.0, 0.6721, 0.0, 0.0, 1.0 , 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,] |
| **precision** | [0.6650, 0.0, 0.0, 0.2822, 0.0, 0.0, 0.2301, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,] |
| **F1-score** | [0.7592, 0.0, 0.0, 0.3975, 0.0, 0.0,0.3742, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0] |
| **支持向量机（SVM）** | |
| **Acc** | 0.9077666666666667 |
| **recall** | [0.9664, 0.3398, 0.9997, 0.9956, 0.9845, 0.9989,0.9927, 0.9583, 0.9885, 0.9999, 0.2537, 0.4762,0.1, 0.2240, 0.9977] |
| **precision** | [0.9179, 0.9792, 0.9931, 0.9968, 0.9763, 0.9793,0.5957, 0.9888, 0.9591, 0.9915, 1.0, 1.0,1.0, 0.7039, 0.8158] |
| **F1-score** | [0.9415, 0.5046, 0.9964, 0.9962, 0.9804, 0.9890,0.7446, 0.9733, 0.9736, 0.9957, 0.4048, 0.6451,0.1818, 0.3398, 0.8976] |
| **KNN（k=5）** | |
| **Acc** | 0.9147166666666666 |
| **recall** | [0.9682, 0.4197, 0.9987, 0.9825, 0.9854, 0.9803,0.9739, 0.9684, 0.9982, 0.9999, 0.6119, 0.57142,0.0, 0.2953, 0.9971] |
| **precision** | [0.9223, 0.9393, 0.9976, 0.9719, 0.9797, 0.9362,0.6275, 0.9718, 0.9593, 0.9981, 0.6119, 0.75,0.0, 0.5714, 0.9905] |
| **F1-score** | [0.9447, 0.5801, 0.9981, 0.9772, 0.9825, 0.9577,0.7632, 0.9701, 0.9783, 0.9989, 0.6119, 0.6486,0.0, 0.3894, 0.9937] |
| **多层感知机(hidden=256)** | |
| **Acc** | 0.7277958333333333 |
| **recall** | [0.9369, 0.4363, 0.5017, 0.1418, 0.0848, 0.3965,0.9023, 0.0013, 0.9963, 0.7650, 0.0, 0.0,0.0, 0.2319, 0.9598] |
| **precision** | [0.7399, 0.7895, 0.9531, 0.8055, 0.2987, 0.6714,0.6003, 1.0, 0.9216, 0.9800, 0.0, 0.0,0.0, 0.2679, 0.8264] |
| **F1-score** | [0.8269, 0.5619, 0.6575, 0.2411, 0.1322, 0.4986,0.7210, 0.0027, 0.9575, 0.8593, 0.0, 0.0,0.0, 0.2486, 0.8881] |

从表4.11可以看到，使用人工特征数据集的传统机器学习算法，最高的准确率是随机森林算法的91.94%，此外，我们也可以从下图4.9的展示的这几个算法的f1-score看着几种算法对各个类别的检测效果。

图4.14 机器学习算法在人工特征集上的F1-score指标表现

Fig. 4.14 F1-score performance of machine learning algorithms on artificial feature sets

从图4.14可以看到，算法svm，随机森林（random forest），多层感知机（mlp）以柱状图的形式展现，而knn和决策树（decision tree）以折线图呈现。毫无疑问，这些算法只有在个别类上的表现可以和本文提出的深度学习模型相比，其他类上的检测远远不如本文4.1节提出的模型。并且在对正常样本的识别上，传统机器学习算法对正常样本识别的F1-score最高只有94%，而我们的深度学习模型有99.96%%。

4.4 本章小结

本章提出了一个基于深度学习的端到端的异常入侵检测模型。该模型包括4个模块，为卷积模块，循环模块，注意力模块，分类模块。端到端的体现是该模型的输入是网络流量字节数据（第三章构建的会话样本集），输出是对样本是否正常的判断。基于该模型，本章做了对比实验，在深度学习实验组中通过逐步调整参数，逐渐提升模型性能，最终本文深度学习模型的准确率为99.96%，另一组实验是使用人工特征集的传统机器学习实验，在实验中这些算法的准确率最高为91.47%。实验证明，本文提出的深度学习模型优于传统方法。

1. 基于全局积分梯度的模型可解释性方法

本章阐述基于全局积分梯度(Integrated Gradients)的模型可解释性方法。

模型可解释性指的是人类对于模型的决策判断机制的理解程度和对模型结果的理解程度的大小。解释模型需要明确模型预测与模型输入特征之间的关系（归因），即模型输入为什么使模型做出这样的预测？模型在输入和输出之间的预测行为是否可以理解？输入的特征那些对决策有着更加重要的影响？

深度学习模型本质上是一个函数，类似于y=f（x）（简单类比）的形式，通过对输入x中某个特征xi 求导，得到y相对于xi的梯度。这个梯度值的大小反映了输入对于输出的贡献度。相当于模型决策时对于那个特征更为看重。因此，梯度常常被用于模型可解释方法中。

而积分梯度是一种通过计算模型输入和基线之间路径上平均梯度来得到归因的一种可解释性方法。基于积分梯度，本文提出一种计算全局积分梯度的方法，本章将根据全局积分梯度法解释第4章所提出的深度模型，得到二维会话图像输入的归因可视化图，以及数据包重要性排名，并对全局积分梯度的结果进行分析总结。

5.1 IG积分梯度

5.1.1 归因定义

归因就是为结果找到原因，从深度神经网络来说，就是对于模型输出，输入中那个元素更加重要？多大程度上贡献了此次预测。

我们定义函数F(x)为一个Rn的映射，代表一个深度神经网络，输入**x**={x1,…, xn}，（这不代表函数的输入只能是向量，对于图像张量输入，可以展开成一维向量，xi只是代表输入的一个元素，一个特征）。归因就是输入**x**和基线后，函数F的预测是多大程度由输入的某个特征造成的一种量化的数学表达。因此定义一个输入**x**经过函数F映射，输出预测后，对输入进行权重分配后得到的向量为该输入情况下的一个归因：AF(x，)=（a1, …，an）, ai就是输入xi对输出预测F（**x**）的贡献，也就是一个权重。

从业界实际运用的两个重要领域来说：在计算机视觉的对象检测领域，输入是彩色图像，而归因也就是对输入的每个元素都给了一个重要性分数，也就是说重要性越高说明这些元素（在图像中就是像素）对于模型做出该区域属于那个对象的预测贡献度高。

在视觉表现上，我们可以将归因可视化，就是一些高亮的像素点，因此如果归因方法正确且符合人类理解的常识，对于一张包含一只鸟的图像，归因方法应该将鸟所在的那些像素高亮显示；在自然语言处理领域：输入是一个个单词组成的句子。归因方法将会为输入句子中的每一个单词分配一个重要性分数，也就是说我们将知道对于自然语言模型，模型做出某种预测是主要由那几个单词造成的。具体来说，在电影评论感情色彩判断中，如果输入影评中出现了“值得一看”，“推荐观看”，等很明显的正面评价，那么一个不错的模型根据该评价认为该影评是正面积极的，一个好的归因方法就应该将这几个单词给予高的分数，即这些单词在所有单词中重要性排名靠前。

5.1.2 两个基本公理

如何评价一个归因方法的好坏呢？作者Mukund Sundararajan等[36]认为好的归因方法需要满足两个公理。

首先声明基线（baseline）的定义：一个固定的输入，用于和其他输入进行对比。通常来说就是一个全0的输入，在图像中就是一个全黑图片，自然语言处理中就是“填充值”。

1. 敏感性

敏感性有两个方面：1：对于输入中任何和基线不同的特征，如果造成了预测不同，那么应该给予该特征非0的贡献度（重要性分数）。2：反过来说，如果模型的预测函数在数学上不依赖于某些特征，对于这些特征应该一直都给予0的贡献度。

在积分梯度之前一个朴素的方法就是根据梯度来给输入分配重要性分数，但是梯度不满足敏感性。举一个具体的例子，假如我们的神经网络是一个简单函数F(x)=x(如果 x<10,否则F(x)=10)，在输入x=10之前是斜线，在x=10之后是一条平坦的F（x）=10的直线。如果使用梯度归因方法，那么对于x是小于10的输入，他会给与重要性分数x，对于x大于10的输入，梯度方法会给与0的重要性分数。然而对比基线0，对于输入x>10，模型的预测是完全不一样的，但是却给与了一样的重要性得分，这就破坏了敏感性。

对于敏感性第2个方面也有解释：如果根据某个归因方法，模型对输入某些特征具有高的重要性得分，如果给这些值扰动（即改变这些特征的值）那么模型在下一次预测中给与这些特征的分数应该大幅下降，然后事实上，模型只是简单的小幅度降低了分数。这个现象表明，这种归因方法没有满足对于不敏感的元素给与一直为0的分数的敏感性原则，该归因方法可能对于一些没有对输出预测做出贡献的特征给予了不该有的重要性分数。

1. 实现不变性

实现不变性：对于功能等效的网络：尽管网络结构不同（函数不一样），但是只要输入相同，则输出相同，归因应该是完全一样的。即将一个输入，分别送入两个不同但是功能等效的网络，输出是一样的（即预测一致），如果一个归因方法满足实现不变性，那么对这两个网络应用归因方法，给出的归因结果应该是一样的（即对于输入的每个特征的重要性得分是一样的。）

首先很多使用离散梯度的归因方法就不满足实现不变性。在梯度反向传播中，加入f是输出，g是输入，h是隐藏层，根据链式规则：

 （5.1）

这也是反向传播的计算方法，但是这是对于连续的区间来说是符合的，对于离散的计算方法是不符合的，如式5.2。

 （5.2）

例如：

因此使用离散梯度的归因方法不满足实现不变性。

是否满足上述两个公理对于一个归因方法的好坏十分重要。因此，从满足这两个公理出发，作者[36]改进了过往方法，提出了积分梯度的归因方法。

5.1.3 IG计算方法

关于积分梯度的一个直觉就是，既然单纯的梯度归因方法只是取了输入在某个值时的梯度，并且这个梯度破坏了敏感性原则，那么如果从将输入x从基线开始按照某个路径增长到x，并且取这个路径的梯度积分作为归因，是不是更加合理?这样计算，规避了之前由于x取在了平坦区，即使它对输出有贡献也因为只取在该点的梯度的原因，最后造成了归因为0。

我们定义i为输入x的某个维度（向量的第几个元素），为输入，为基线，为比例，介于[0,1]之间，代表插值的输入多大程度上接近输入x。

因此对于输入x在第i个维度的积分梯度计算公式定义如下：

 （5.3）

式中，为模型预测函数F对输入x在特征i处的梯度。

从式5.3可以看到模型预测函数F的输入是：，就是一个从基线开始取比例的的一个中间状态x，即插值输入。

可以证明，上述的积分梯度计算方式满足完整性：

 （5.4）

完整性：积分梯度在各个特征上的和等于预测函数F在x和基线的值的差。完整性是敏感性的一个强力的证明，因此积分梯度满足敏感性。这其实非常容易理解，如果积分梯度不满足敏感性，则积分梯度会对预测函数不依赖的特征分配一个非0的重要性分数，因为预测函数的值完全不依赖这个特征，那么给这个无关特征分配了重要性分数，那么积分梯度在个特征维度上的和将不等于F在输入x和基线之间的差值。

对于路径，即如何从基线到达x，有多种方式，最简单直接的方式就是直线路径。采用直线路径由两个优点，其一是计算简单，其二是对称性。对称性：如果输入中两个特征变量相互交换位置不改变输出，那么归因方法给予者两个变量完全一样的重要性得分。

由于直接计算积分梯度往往不可行，在实际计算中，我们采用如下黎曼近似计算积分梯度：使用直线路径从到达x，并且用有限的步数m来模拟这个路径，则积分梯度的近似值为式（5.5）[36]：

 （5.5）

式中，m就是计算积分梯度中黎曼近似的步数。对于该值，取20-300就可以在小误差范围内得到积分梯度的近似值。

因此，计算积分梯度的详细步骤如下：

1.选择基线：在第4章中本文提出的深度学习模型输入为：32\*32的黑白图像，以及320维的一维数据包序列。基线的选择是和输入形成对比，理论上来说，基线就是背景，是坐标系的零点。因此，从实践和直觉出发，全0的输入应该是不错的基线。因此，基线是全0输入的32\*32的二维图像矩阵和全0的320维一维输入向量。

2.步数选择：采用黎曼近似的方法求解积分梯度的近似值。步数m代表对路径上点的个数的选取，步数越大，路径上的点越多，近似解的精度也越高，但是计算的代价也会相应的增加。本文算法采用直线路径，步数m取100。

3.计算：将基线和步数m代入公式5.5，求得该次预测对输入某个特征i的积分梯度。结果与输入形状一致，即一个32\*32的二维矩阵，一个32\*320的矩阵。

5.2 可解释方法实验

深度学习模型本质上是一个有着极大参数的函数。在单个预测后，模型输出结果，以该结果为具体值，可以求得模型预测函数对输入某个特征得梯度。而梯度反映了函数变化的快慢，梯度大则模型输出对于输入的微小改变就变化的大，即该特征对模型输出的影响大，贡献度高。

而积分梯度就是通过求模型输入x到基线这个变化过程中直线路径上点的梯度的平均值。梯度通过反映输入对输出的贡献度而被用于解释模型，理解模型的判断机制，决策函数。

根据积分梯度的结果我们可以得知输入中那个特征更为重要，但是每次结果都是单个样本的结果，不是全局性的特征重要性，为此需要做出改进，为此提出计算全局积分梯度的方法。

实验目的：通过改进的称为全局积分梯度的归因方法，理解模型决策机制。

实验步骤：

1. 计算全局积分梯度，将所有会话样本输入，计算每个类别的全局积分梯度。所得结果是和模型输入形状一致的两个张量，为32\*320的张量和32\*32的矩阵。
2. 获得该张量我们需要计算数据包重要性排名：将32\*320的第二维度加和得到一个32维的向量，对比这32个数值的大小，就能得出数据包重要性排名，
3. 取最重要的数据包的归因结果，为1\*320的向量，尝试匹配数据包格式关键位，论证相关性，在5.3.2结果理解中。
4. 32\*32的矩阵，将其可视化，则可以得到可视化的归因图，在可视化图中，颜色越深的像素点全局积分梯度就更大，代表对输出的影响越大，贡献度越高。该结果会在结果分析中5.3.1展示。

5.2.1 全局积分梯度

全局的特征重要性需要全局的样本来参与。由于无法得到真实的数据来计算，但是本身深度学习就是从训练集习得知识的，使用样本集进行近似求解也足够。

全局积分梯度的定义非常简单，如式5.6：

 （5.6）

式中，c代表类别，i为特征维度，k是c类样本数目。

我们需要分别计算每类中的全局积分梯度，在样本集中，如果类别c的样本数量为k，则需要分别计算该类中每个样本的积分梯度，然后求均值。

需要指出的是，这种根据训练集中某个类别所有样本而计算的全局积分梯度往往需要根据模型任务进行一定的改变。例如，在对象识别领域，某个类对应的是某个区域，但是该区域有大有小，且其中的对象展露的“面”不一致，如果只是简单对整个图像进行全局积分梯度的计算，没有任何意义，且无法理解。正确的思路应该是这样：例如我们想获得模型对猫的全局积分梯度，就应该先将猫进行分解：化为头，躯干，四肢，尾巴等，也可以更加细致的划分。然后提取该区域狗的积分梯度，分别统计各个部分的重要性分数总和，然后累加到各个区域上（要除以像素个数，有的猫分辨率太小），最后除以样本数，如此就能得到模型对猫的输入的全局归因分布。

思路的转变关键是归因方法要找到模型的那些输入导致了模型输出，而不是固定的输入中的某个位置。在对计算猫的全局积分梯度时，就体现了这样的思想，我们输入了不同的猫，模型对猫的什么样的特征敏感（头？尾巴？或者总体形状？），这样才是可理解的归因方法，同时也证明一个好的模型在做人类理解任务上对输入的归因应该符合直觉。

另一个领域是自然语言处理，例如对影评积极或者负面的任务中，应该按如下方法计算全局积分梯度：计算单个样本时，记录下每个单词的积分梯度（有时需要统计词语和短语的积分梯度），在所有某个类别的样本计算完之后，将属于某个单词的积分梯度相加，除以出现的次数，就得到了该单词或词组短语的全局积分梯度。符合人类理解的解读就是，如“好看，精彩，吸引人”等词语毫无疑问会在模型对影评做出积极评价中获得高分。

最后，回到本文模型。本文模型的输入具有特殊性，本文模型的输入网络流量数据，是原始字节流。首先，网络流量数据遵循固定格式，即TCP/IP协议，因此在本文计算全局积分梯度时可以简单的求均值尽心计算，而不需要一些比较复杂的转换处理。当然，协议规定的只是头部格式，实际承载的数据内容是没有固定格式的，因此该方法计算的全局积分梯度也具有一定局限性。但是，如果异常样本的内容数据格式遵循一定的规律性，也是具有意义的。

5.2.2 数据包重要性排名

我们的模型输入中其一是一系列数据包，可以类比自然语言处理中的一句话（是由一个个单词组成），计算该积分梯度，就相当于计算模型将输出归因到这一句话中的各个单词中，给出对输出的重要性分数，即可以对数据包进行排序，按照归因方法得到的重要性分数进行从大到小的排序。

数据包是网络上主机交流的语言，但是自然语言交流中每一个单词都具有具体意义，归因后人类是可以理解的，对于数据包还是需要一个解释。对于具体协议TCP来说，建立连接的前3个数据包简化成打招呼的数据包，而之后的数据包的具体意义则多种多样，完全根据使用者的意图来确定，需要具体问题具体分析。比如，对于僵尸网络的会话样本中的一系列数据包，可能除了之前和服务器端的打招呼后，之后的交流就只是单方面给僵尸网络服务器端传送被倾入主机的隐私数据，或者接受某条指令，执行后门程序等。需要单个样本具体分析。

我们不需要如此详细的做这些繁琐的工作，而是简单的根据数据包重要性排名，对给予高重要性分数的数据包重点关注。当然这一切都需要建立在模型判断的依据和人类理解任务一致。如对象识别，就因该将整个猫的轮廓凸显出来（人类的形状判断），但是在本文中，可能模型的判断依据不是这些高级抽象特征，而是因为数据形式展现的某些纹理特征，如在第3章中对会话样本的可视化中，就包含明显的图形特征。但是网络流量数据展现的图形特征是否能够具有高级抽象还需要在后续的结果分析中进一步判定。

5.2.3 数据包格式关键位

模型的输入在是遵循网络协议的网络流量数据，对于首部格式，我们可以确定每个位置的意义，因此从积分梯度得到的重要性分数，可以对应到首部格式的各个值上。

该分析是一种假设。我们不能确定模型给出的重要性分数和对应位置的格式定义一定是相关联的，只能提供一种参考，尝试发掘出一些有用的知识关联。

以太网数据帧格式：



图5.1 以太网数据帧格式

Fig. 5.1 Ethernet data frame format

IPv4数据报格式：

图5.2 IPv4数据报格式

Fig. 5.2 IPv4 data frame format

TCP数据包格式：

图5.3 TCP数据报格式

Fig. 5.3 TCP data frame format

UDP数据报格式：源端口（2），目的端口（2），长度（2），校验值（2），数据。

以上从图5.1，5.2，5.3分别展示了目前互联网基础的几种协议的数据格式，为之后的分析提供对应关系。

5.3 可解释方法实验结果分析

在本节中，我们将根据全局积分梯度进行模型理解分析，从可视化图和数据包重要性排名，这些抽象的数据结果中归纳出模型的判断决策机制，理解模型。

5.3.1 结果可视化图

1. FTP-BruteForce和SSH-Bruteforce

图5.4为FTP-BruteForce攻击类中，输入为会话图像部分的全局积分梯度可视化图。

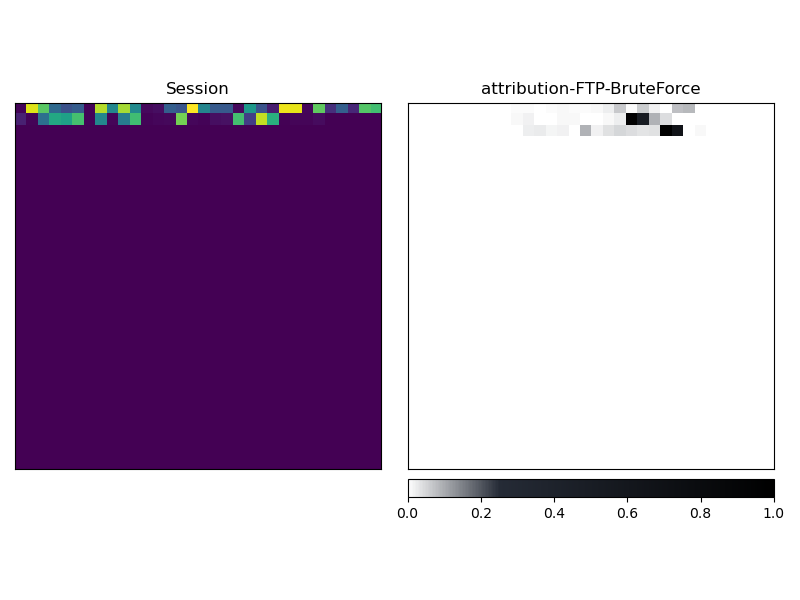


图5.4 FTP-BruteForce全局积分梯度可视化图

Fig. 5.4 Global IG Visualization of FTP-BruteForce

FTP-BruteForce数据包重要性排名：[1,2]。其中p1是第一个数据包，p2是第二个数据包，重要性分数为：[0.9765,0.2150],可见在对该攻击类型的检测中，第一个数据包对模型预测最为重要。

图5.5为SSH-Bruteforce样本类中会话图像部分的全局积分梯度图。

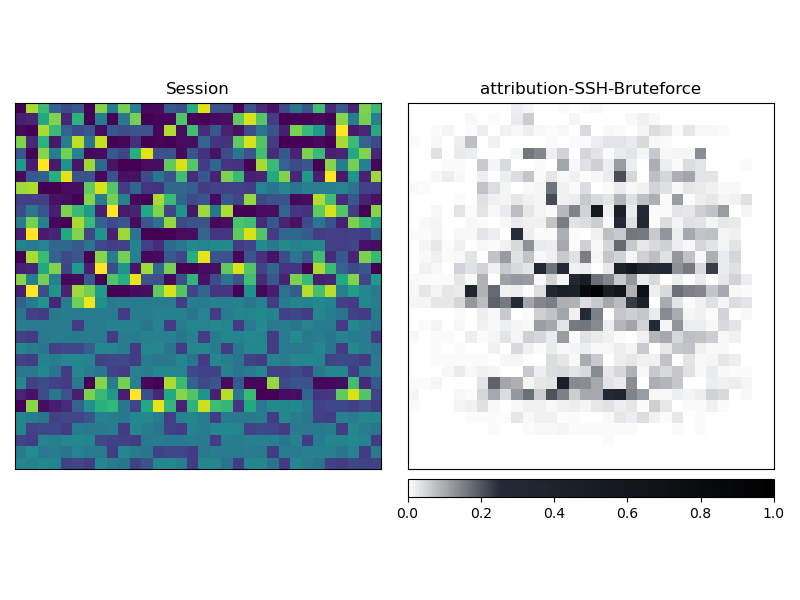


图5.5 SSH-Bruteforce全局积分梯度可视化图

Fig. 5.5 Global Integral Gradient Visualization of SSH-Bruteforce

数据包重要性排名：[ 1, 2, 3, 31, 4, 15, 14, 17, 32, 18, 16, 5, 19, 7, 12, 9, 6, 10,21, 30, 22, 20, 28, 25, 27, 24, 13, 23, 29, 26, 8, 11],只从前7个数据包看，模型对前4个数据包1，2，3，4和位于尾部一个数据包31，以及中部的数据包14，17给予了高的分数，即对于SSH-Bruteforce的攻击检测，输入的首部和中部，尾部对输出预测有着重要贡献。

1. DoS-GoldenEye 和DoS-Slowloris

图5.6所示，展示了样本类别为DoS-GoldenEye的图像输入部分的全局积分梯度可视化图。

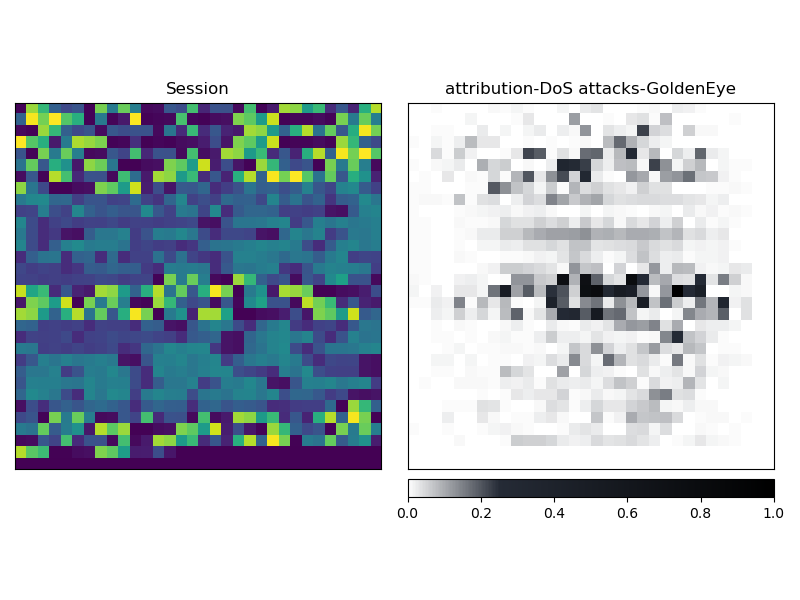


图5.6 DoS-GoldenEye全局积分梯度可视化图

Fig. 5.6 Global Integral Gradient Visualization of DoS-GoldenEye

数据包重要性排名：[ 1, 6, 4, 3, 2, 7, 8，5]。从排名可以看到，第一个数据包仍然十分重要，但是位于整个输入的第6个数据包也非常关键。即首部和中部对输出贡献较大。

图5.7所示，展示了类别DoS- Slowloris的全局积分梯度可视化图。

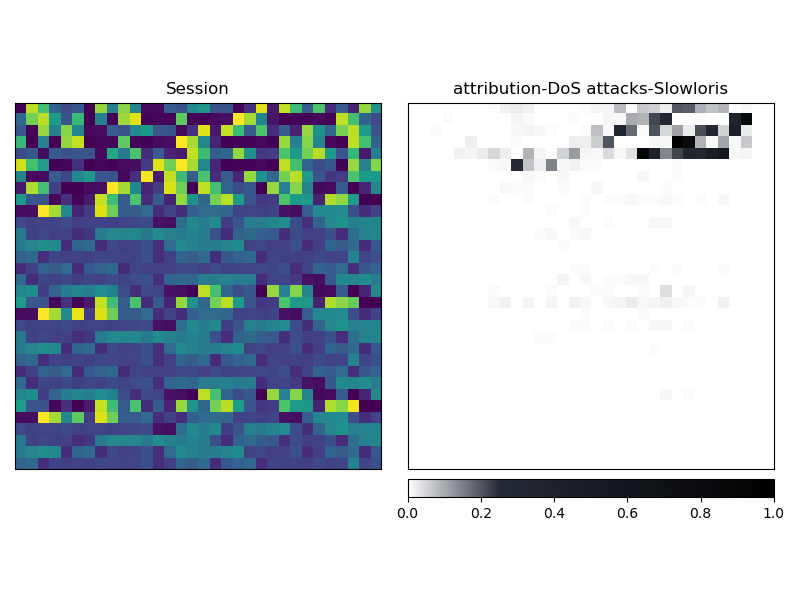
****

图5.7 DoS-Slowloris全局积分梯度可视化图

Fig. 5.7 Global Integral Gradient Visualization of DoS-Slowloris

数据包重要性排名：[ 1, 2, 3, 17, 15, 14, 18, 4, 16, 13, 11, 7, 8, 10, 9, 6,12, 5]。可知，第一个数据包得分最高，其次是第2，和3个，但是位于尾部的第17个数据包也得到了比较高的分数。输入中的首部和尾部对输出贡献较大。

1. DoS Hulk和 DoS SlowHTTPTest

图5.8所示，展示了类别DoS Hulk的全局积分梯度可视化图。

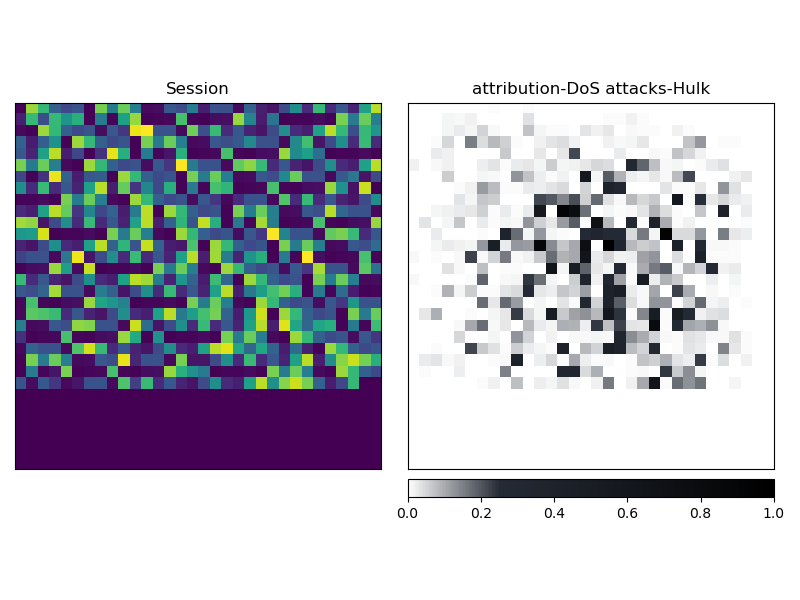


图5.8 DoS Hulk全局积分梯度可视化图

Fig. 5.8 Global Integral Gradient Visualization of DoS Hulk

数据包重要性排名：[ 1, 14, 2, 16, 3, 5, 4, 8, 6, 7, 9, 11, 10, 12, 13, 15, 17]。第一个数据包和第14个数据包，得分最高的数据包位于输入的首部和尾部。

图5.9所示，展示了类别DoS SlowHTTPTest的全局积分梯度可视化图。

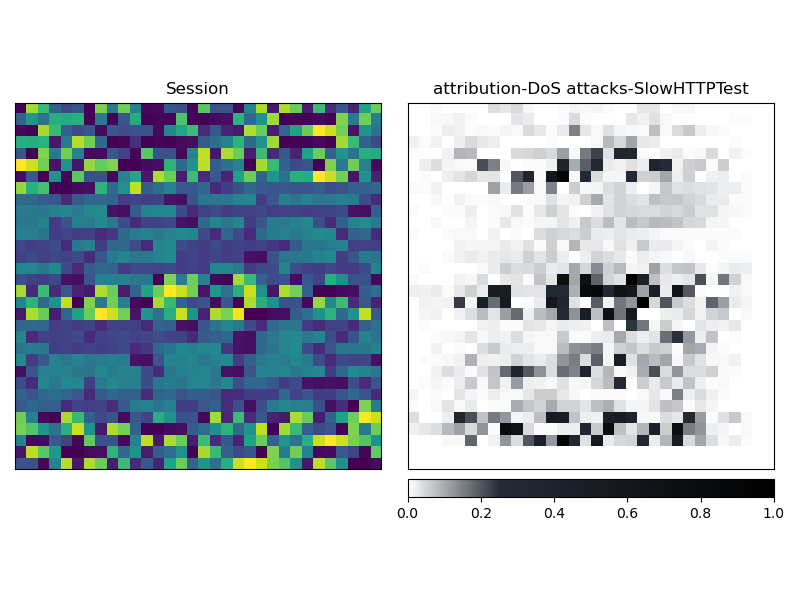


图5.9 DoS SlowHTTPTest全局积分梯度可视化图

Fig. 5.9 Global Integral Gradient Visualization of DoS SlowHTTPTest

数据包重要性排名：[ 1, 4, 2, 32, 6, 3, 8, 10, 28, 9, 11, 26, 13, 27, 5, 18, 12, 21,23, 29, 31, 17, 7, 22, 20, 14, 19, 25, 15, 24, 16, 30]。可知，得分较高的数据包位于输入的首部和尾部，并且第一个数据包重要性分数第一。

1. DDoS LOIC-HTTP

图5.10所示，展示了类别DoS LOIC-HTTP的全局积分梯度可视化图。

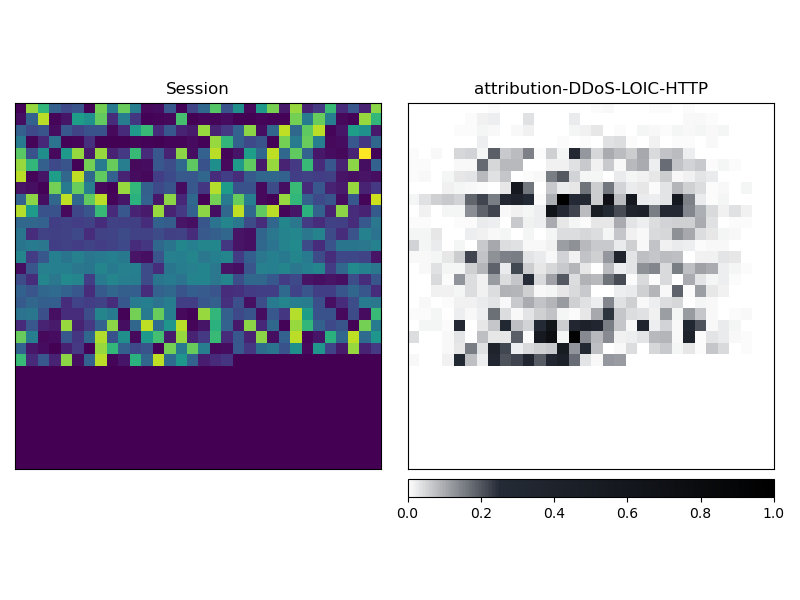


图5.10 DoS LOIC-HTTP全局积分梯度可视化图

Fig. 5.10 Global Integral Gradient Visualization of DoS LOIC-HTTP

数据包重要性排名：[ 1, 6, 9, 2, 4, 5, 8, 3, 7, 10, 11]。可知，输入中对输出由较高贡献的数据包是第一个，第6个，位于输入的首部和中部。

1. DDoS LOIC-UDP和DDoS HOIC

图5.11所示，展示了类别DDoS LOIC-UDP的全局积分梯度可视化图。

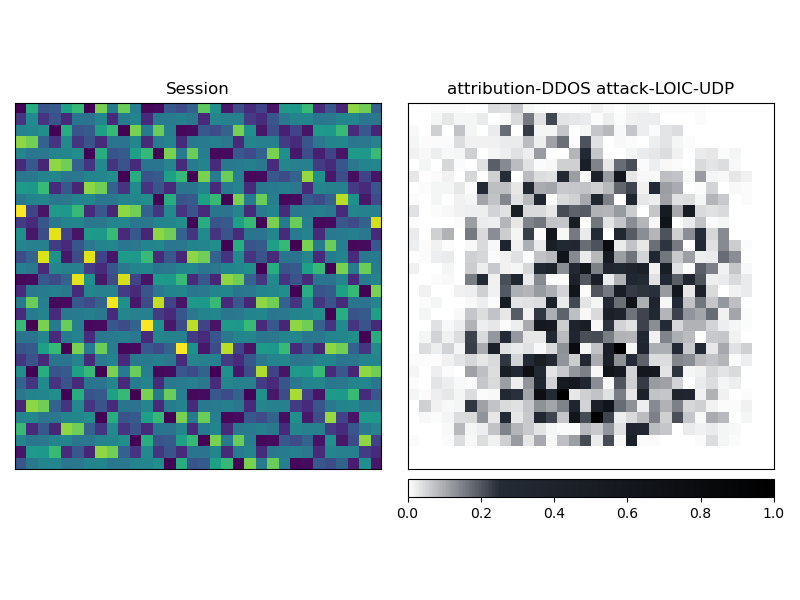


图5.11 DDoS LOIC-UDP全局积分梯度可视化图

Fig. 5.11 Global Integral Gradient Visualization of DDoS LOIC-UDP

数据包重要性排名：[ 1, 2, 32, 3, 4, 31, 5, 6, 7, 8, 9, 29, 28, 10, 27, 11, 26, 30,12, 25, 24, 13, 23, 14, 22, 15, 21, 16, 20, 17, 19, 18]。此次的攻击类型比较特殊，因为该类会话样本中的每一个数据包是一样的，为典型的流量攻击，以巨大的带宽消耗让目标服务器不能响应。在数据包排名中，虽然可以看到前几名是第1，2个和最后一个数据包，但是除了第一个数据包之外，其他数据包的重要性得分相差非常小，可以认为其他数据包对模型输出的贡献几乎一致。这与我们的逻辑判断近乎一致，因此可以认为模型的可理解性在该类攻击上和人类达成了一致。

图5.12所示，展示了类别DDoS HOIC的全局积分梯度可视化图。

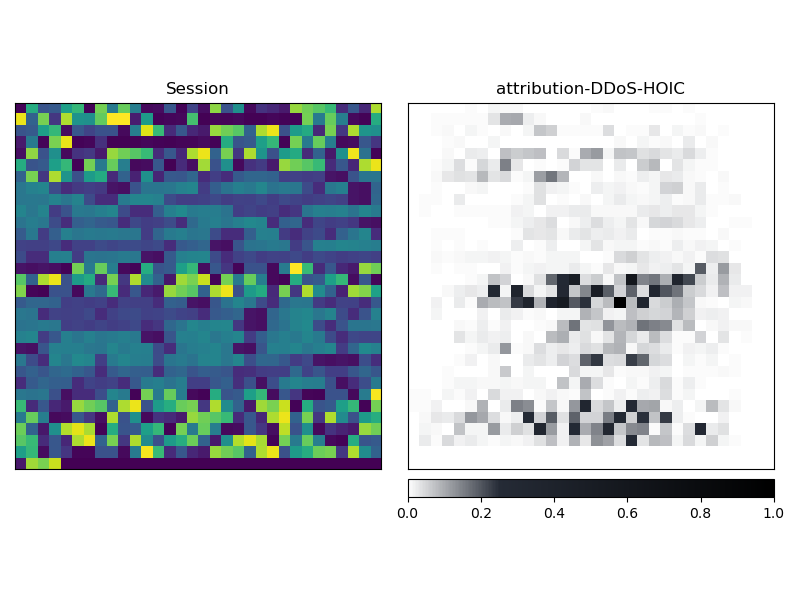


图5.12 DDoS HOIC全局积分梯度可视化图

Fig. 5.12 Global Integral Gradient Visualization of DDoS HOIC

数据包重要性排名：[ 1, 2, 10, 9, 6, 3, 7, 4, 8, 5, 11]。可知，第一个数据包得分最高，其次是第2个和第10个，因此，输入中首部尾部的数据包对模型输出贡献较大。

1. Brute Force -Web和Brute Force -XSS和SQL Injection

图5.13所示，展示了类别Brute Force -Web的全局积分梯度可视化图。

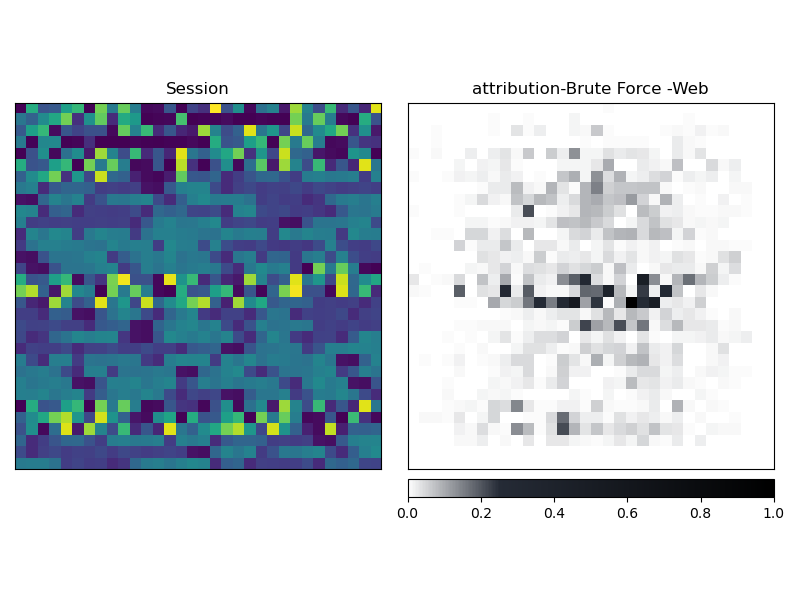


图5.13 Brute Force -Web全局积分梯度可视化图

Fig. 5.13 Global Integral Gradient Visualization of Brute Force -Web

数据包重要性排名：[ 1, 2, 3, 4, 32, 6, 10, 31, 11, 8, 7, 9, 12, 27, 22, 17, 28, 18,23, 13, 26, 21, 16, 15, 20, 19, 14, 24, 25, 29, 5, 30]。可知，首部尾部对输出贡献较大，第一位的仍然是第一个数据包。

图5.14所示，展示了类别Brute Force -XSS的全局积分梯度可视化图。

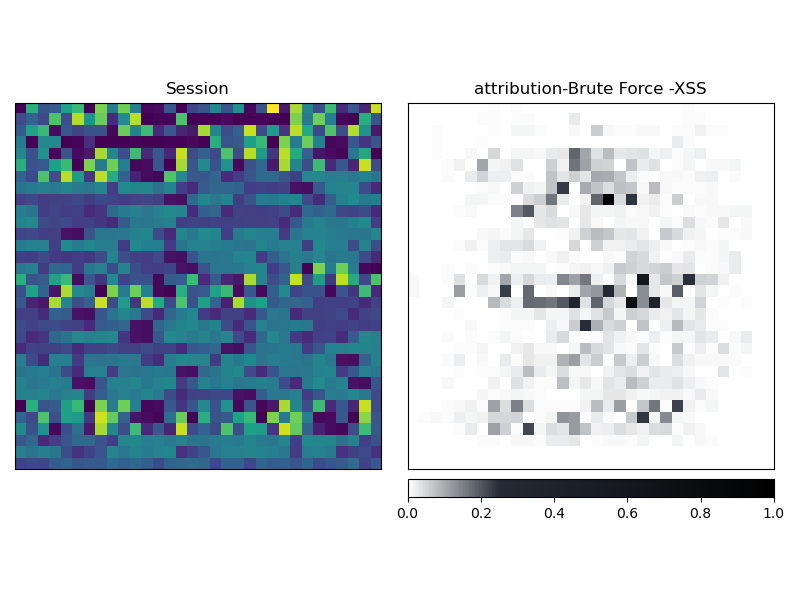


图5.14 Brute Force -XSS全局积分梯度可视化图

Fig. 5.14 Global Integral Gradient Visualization of Brute Force -XSS

数据包重要性排名：[ 1, 2, 3, 4, 31, 6, 8, 32, 11, 13, 28, 16, 26, 22, 25, 19, 10, 17,20, 14, 23, 29, 30, 7, 21, 18, 9, 12, 24, 15, 27, 5]。可知，数据包贡献第一名是第一个数据包，对模型输出贡献较大的是首部和尾部的数据包。

图5.15所示，展示了类别SQL Injection的全局积分梯度可视化图。

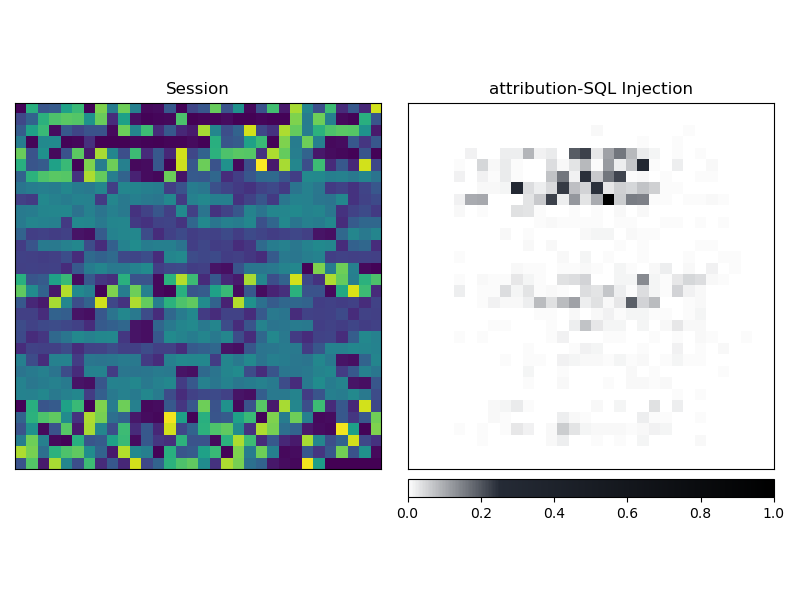


图5.15 SQL Injection全局积分梯度可视化图

Fig. 5.15 Global Integral Gradient Visualization of Brute SQL Injection

数据包重要性排名：[ 1, 2, 6, 3, 4, 10, 11, 8, 9, 7, 5]。可知，输入中前几名基本是前几个数据包，因此输入中对模型预测贡献较大的数据包时位于首部的数据包。

1. Infilteration

图5.16所示，展示了类别Infilteration的全局积分梯度可视化图。

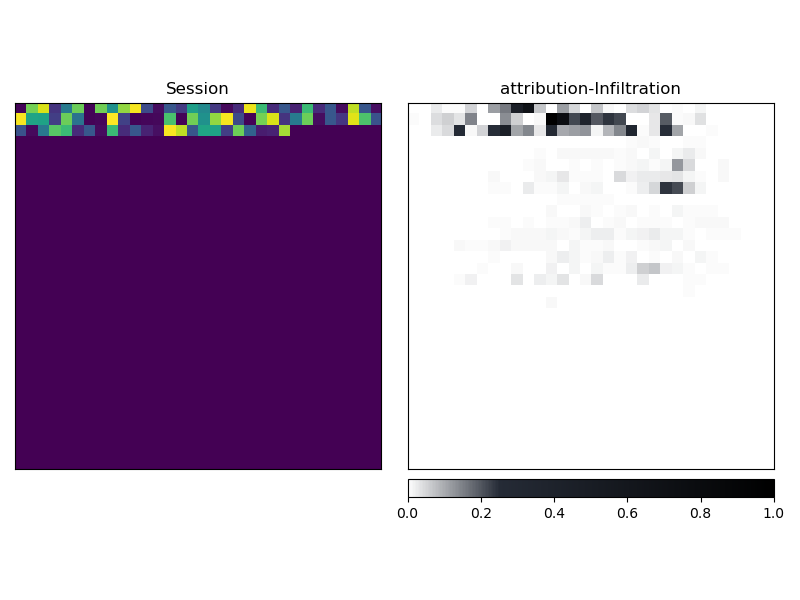


图5.16 Infilteration全局积分梯度可视化图

Fig. 5.16 Global Integral Gradient Visualization of Infilteration

数据包重要性排名：[ 1, 2]。可知，渗透攻击数据包较少，重要性得分最高的数据包就是前两个数据包，第一个数据包依然十分重要。

1. Bot

图5.17所示，展示了类别Bot的全局积分梯度可视化图。

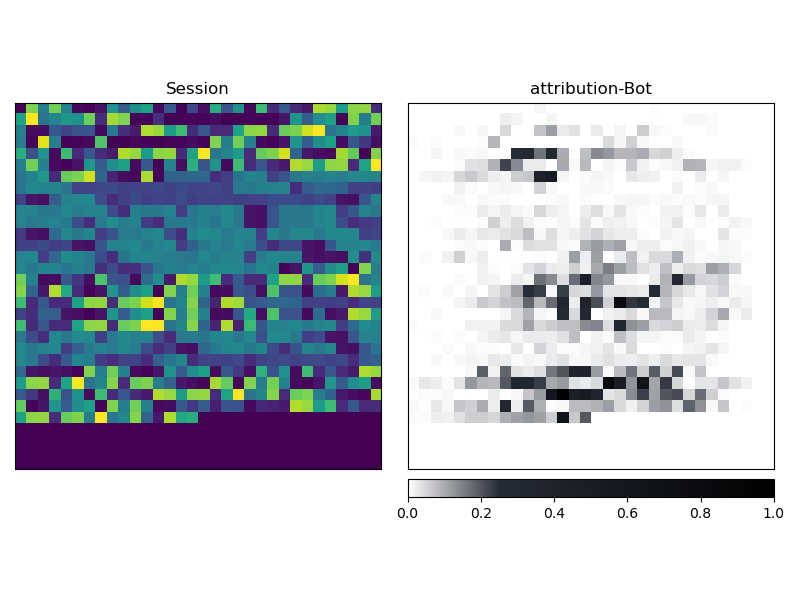


图5.17 Bot全局积分梯度可视化图

Fig. 5.17 Global Integral Gradient Visualization of Bot

数据包重要性排名：[ 1, 10, 4, 7, 6, 2, 8, 3, 9, 5]。重要性得分前两名是第一个数据包和最后一个数据包。僵尸网络中，僵尸客户端和服务器端通信联系，重要信息可能在最后发出，发出后就断开连接，等待下一次通信。可知。模型输入对模型输出有较大贡献的数据包位于输入流的首部和尾部。

5.3.2 结果理解

在上一节中，我们展示了全局积分梯度的结果。从中我们可以得到两个重要的结论：

一，模型根据会话图像纹理特征进行检测。在模型输入中的会话二维图像的全局积分梯度可视画图中，可以看到输入图像的纹理特征得到了较高分数（黑色部分）。因此，可以判断模型在对攻击进行检测之时，对于会话图像部分的是利用纹理特征，根据各种攻击类型会话图像不同的纹理特征来识别它们。

二，第一个数据包是最重要的。对于模型的另一个输入部分：一系列数据包。根据全局积分梯度，我们可以给出数据包重要性排名，其中有一个固定不变的点：第一个数据包重要性分数在所有类别中都位居第一。

此外，我们在第5.2.3中提出了数据包格式关键位的猜想，即根据重要性分数来对应数据报协议中的具体定义。以DDoS-LOIC-UDP为例，我们使用会话输入中的一系列数据包中第一个数据包的重要性分数进行分析：

排名前5的几个字节是：26，6，28，2，48。该攻击的类别中，数据包的格式是：以太网（14），ip（20），udp（8）。因此，其中第26个字节对应的是udp的前两个字节，即源端口。第6字节：源mac地址，第28字节：目的端口，第2字节：源mac地址，第48字节：udp数据部分。

对于udp洪泛攻击，攻击者最重要的特点是极大的流量来消耗服务器资源，通常表现为在短时间的巨大异常流量，对于该特点，本文模型中有一定体现（即该攻击会话样本是32个数据包组成，通常其他样本数据包较少，50%的样本只有10个以下数据包，详情见3.3.2）。在对DDoS-LOIC-UDP数据包格式关键位的分析中，可以看到，模型对于源端口，源mac地址给予了较大的重视，但是对于攻击者来说，这两者其实是可以随意更改的。更重要的点是攻击者要攻击的目的端口，目的ip等信息。值得一提的是目的ip在重要性分数中也有较高排名，只是不在前5名。

以上分析很明确的表现了一个点，那就是模型的重要性分数对应到数据包格式关键位上，并没有非常强的关联，或者说像是人类专家专门分析每种攻击的数据报格式特点，并且提取相应的特征那样明确。

5.4 本章小结

在本章中，我们尝试解释模型。首先，在5.1中介绍了积分梯度的归因方法，这是一种通过计算输入和基线之间直线路径上固定步数的点的梯度的平均值来得到归因的方法。然后，针对该方法，本文提出全局积分梯度的思路来将单个样本的归因扩展到全局样本。最后，通过该全局积分梯度，我们得到了模型的归因可视化结果：包括模型输入中会话二维图像的可视化归因（模型视图通过图像的纹理特征来识别异常），数据包的重要性排名（组成会话样本的系一列数据包中根据归因重要性分数组排序）：分析出了每类攻击中，输入的数据包的对输出做出重大贡献的数据包位置。

在5.2.3模型理解中，本文通过进一步分析，使用全局积分梯度的得到的可解释性结果来论证模型的优缺点，并指明单个数据包中每个字节的重要性分数与数据报格式上相应字节没有强关联，这是模型的整体框架决定的。

1. 总结与展望

6.1 总结

本文的研究内容是怎么使用原始网络流量数据做入侵检测。围绕该主题，论文做了以下工作：首先是数据，数据的来源是CICIDS-2018提供的原始网络捕获pacp文件，该数据是在一个虚拟的公司子网络结构上通过模拟正常用户行为和攻击行为收集而来。对于数据，本文提出一种流量分割方法，构建了一个存储原始网络字节数据的会话样本数据集，该数据集有两个特点：单个数据包截取前320个字节，一系列数据包组成一个会话样本，该会话样本最多含有32个数据包。该数据集包含正常会话样本和异常会话样本在内15种类别，共计80万个样本数，详情见第3章。

其次是入侵检测模型。该入侵检测模型的思路是将会话样本视为两种不同的数据形式输入模型：融合一系列数据包为会话二维图像送入卷积模块，一系列数据包视为一维序列送入循环网络模块。根据卷积模块的输入和循环模块的输入应用注意力模块，最后送入分类器中执行异常入侵检测任务。将网络流量会话样本视为不同数据形式的原因是：融合多个数据包为二维图像可以提取多个数据包形成的图像特征，将系列数据包为一维序列送入循环模块可以从数据包流中提取到两个机器之间对话的序列特征，并且得到表示整个数据包序列的融合特征向量。注意力模块可以通过对会话二维图像不同区域给与不同的权重来更好的专注于特定类别的判断中。在第4章中，本文详细介绍了该模型，并做了详细的对比实验，通过与使用人工特征的实验组进行对比，本文所采用的使用原始网络流量数据的端到端的深度学习模型在异常入侵检测任务的各项性能指标结果中好于人工特征实验组。因此，可以认为该模型有效且具有实用性。

最后，在论文第5章本文介绍了一种使用全局积分梯度的办法来进行模型可解释性工作。该方法基于积分梯度，但是在计算上通过计算全局样本的积分梯度来的得到全局的归因分布（给每个输入特征一个重要性分数）。在5.2.1中详细论述了如何计算全局积分梯度，给出了扩展到不同模型的建议方法。根据全局积分梯度，我么可以得到模型在二维会话图像中重要像素点，以及一系列数据包的重要性排名，根据结果在5.3.2中我们得出关于该模型的重要结论，并论证了将重要性分数对应到数据包格式关键位上的弱相关性。

6.2 展望

深度学习的理论技术发展迅速，有研究者使用图神经网络的深度学习技术进行网络异常入侵检测。图的结构可以将异常入侵检测的初始状态从两个主机之间产生对话扩展到整个网络之间的交流。最理想的状态是将节点表示为主机，边视为节点之间的通信交流，节点可以有节点状态向量，边也可以通过特征向量形式表示通信，将某个时刻或某个时间段之间网络作为输入，通过图神经网络可以执行多个任务：整体网络状态，单个节点状态，边的类别预测（判断连接是否异常）。该设想十分有前景，首要的难点在于数据。回到本文入侵检测模型，有一些进一步的工作可以实施，如：融合人工规则检测系统联合检测，进一步扩大会话样本单个数据包最大字节和数据包数目。增加语义处理模块，将头部格式数据以自然语言处理的方式输入模型，等一些可能方向的验证和实验。当然最为重要的是，入侵检测系统就如现实世界中的安防系统一样，详细的数据形式是更多的监控，而逻辑判断系统则是一个经验丰富而聪明的人，更多的数据，更加丰富的数据特征是一切的关键，在当前，在固定的数据集上验证模型，就相当于在给定相同线索情况下请侦探查出真相。好的侦探固然重要，但是能够从原始数据中探索更多信息或许是更为重要的能力。因此，从该结论上来说，本文模型探索了数据的不同信息形式，获得额外的特征来加强模型的检测能力。

参考文献

[1] 2020 年我国互联网网络安全态势综述[R].北京：国家计算机网络应急技术处理协调中心，2021.

[2] Cisco. SNORT3 [EB/OL]. https://www.snort.org, 1998/2021-11-18.

[3] Bilge Leyla., Dumitraş Tudor. Before we knew it: anempirical study of zero-day attacks in the real world [A]. CCS '12: Proceedings of the 2012 ACM conference on Computer and communications security [C]. New York, Association for Computing Machinery,2012. 833-844.

[4] Chandola V., Banerjee A., Kumar V. Anomaly detection: A survey. (2009) ACM Computing Surveys, 41(3), DOI:10.1145/1541880.1541882.

[5] University of California, Irvine. KDD Cup 1999 Data [EB/OL]. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html,1999.

[6] Scikits learn developer. Scikit learn [EB/OL]. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch\_kddcup99.html,2021.11.18.

[7] Ring M, Wunderlich S., Scheuring D, et al. A survey of network-based intrusion detection data sets [J]. Computers and Security, 2019, 86:147-167.

[8] Mukkamala S., Janoski G., Sung A. Intrusion detection using neural networks and support vector machines [A]. Proceedings of the 2002 International Joint Conference on Neural Networks [C]. New York: IEEE,2002. (2)1702–1707.

[9] Lin Weichao, Ke Shihwen, Tsai Chihfong. CANN: An intrusion detection system based on combining cluster centers and nearest neighbors [J]. Knowledge-Based Systems, 2015,78:13-21.

[10] Hu Weiming, Hu Wei, Maybank Steve. AdaBoost-Based Algorithm for Network Intrusion Detection [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2008, 38(2): 577-583.

[11] Zhang J., Zulkernine M. and Haque A. Random-Forests-Based Network Intrusion Detection Systems [J]. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2008,38(5):649-659,

[12] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural works [J]. Advances in Neural Information Processing Systems.2012,25(2):1097-1105.

[13] Vinayakumar R., Soman K. P., Poornachandran Prabaharan. Applying convolutional neural network for network intrusion detection [A]. 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) [C]. New York: IEEE, 2017.1222-1228.

[14] Yin Chuanlong, Zhu Yuefei, Fei Jinlong, He Xinzheng. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks [J]. IEEE Access. vol. 2017, 5:21954-21961.

[15] Xu Congyuan, Shen Jizhong, Du Xin, Zhang Fan. An Intrusion Detection System Using a Deep Neural Network with Gated Recurrent Units [J]. IEEE Access, 2018, 6: 48697-48707.

[16] Wang Wei, Zhu Ming, Zeng Xuewen, Ye Xiaozhou，Sheng Yiqiang. Malware traffic classification using convolutional neural network for representation learning [A]. 2017 International Conference on Information Networking (ICOIN) [C]. New York: IEEE,2017.712–717.

[17] Wang Wei, Zhu Ming, Wang Jinlin, Zeng Xuewen, Yang Zhongzhen. End-to-end encrypted traffic classification with one-dimensional convolution neural networks[A]. 2017 IEEE International Conference on Intelligence and Security Informatics (ISI) [C]. New York, IEEE,2017.43-48.

[18] David Pujol-Perich, José Suárez-Varela, Albert Cabellos-Aparicio, Pere Barlet-Ros. Unveiling the potential of Graph Neural Networks for robust Intrusion Detection [EB/OL]. https://arxiv.org/pdf/2107.14756.pdf，2017/2021-09-22.

[19] Cheng Qiumei, Wu Qiumei, Zhou Shiying. Discovering Attack Scenarios via Intrusion Alert Correlation Using Graph Convolutional Networks [J]. IEEE Communications Letters, 2021, 25(5): 1564-1567.

[20] Yang Yunyi, Wang Liming. LGANet: Local Graph Attention Network for Peer-to-Peer Botnet Detection [A]. 2021 3rd International Conference on Advances in Computer Technology, Information Science and Communication (CTISC) [C]. New York, IEEE ,2021. 31-36.

[21] Garcia S., Grill M., Stiborek J.，Zunino A. An empirical comparison of botnet detection methods [J]. Computers and Security, 2014,45: 100-123.

[22] Brazdil P, Giraud-Carrier C, Soares C, Vilalta R. Meta-Learning - Concepts and Techniques [M]. Data Mining and Knowledge Discovery Handbook, Boston, Springer,2010.717-731.

[23] Zoppi Tommaso, Ceccarelli Andrea. Prepare for Trouble and Make it Double! Supervised – Unsupervised Stacking for Anomaly-Based Intrusion Detection [J]. Journal of Network and Computer Applications, 2021, 189(3)，103106.

[24] Quinlan J.R. C4. 5: programs for machine learning. [M]. San Mateo, 1993, Morgan Kaufmann Publishers.

[25] Breiman L, Friedman J, Olshen R, Stone C. Classification and Regression Trees [EB/OL]. https://ocw.mit.edu/courses/sloan-school-of-management/15-062-data-mining-spring-2003/lecture-notes/L3ClassTrees.pdf,1984/2021.

[26] Burges, C.J. A Tutorial on Support Vector Machines for Pattern Recognition [J]. Data Mining and Knowledge Discovery,1998, 2: 121–167.

[27] Zhang Harry. The optimality of Naive Bayes [A]. Conference: Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference [C]. Florida, FLAIRS, 2004(2):562-567.

[28] Breiman Leo. Random Forests [J]. Machine Learning,2001, 45(1): 5–32.

[29] Breiman Leo, Cutler Adele. A Brief Overview of Random Forests and Some Comments about The Features of the Method [EB/OL]. https://www.stat.berkeley.edu/~breiman/RandomForests/cc\_home.htm#overview, 2001/2021-11-18.

[30] Freund Yoav, Robert E.Schapire. Experiments with a New Boosting Algorithm [A].Machine Learning: Proceedings of the Thirteenth International Conference [C], Italy, Morgan Kaufmann,1996:148-156.

[31] Lecun Y., Bottou L., Bengio Y., Haffner P.. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[32] Cho Kyunghyun, Merrienboer B.V, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation [EB/OL]. https://arxiv.org/abs/1406.1078v3,2014-06-03/2021-11-19.

[33] Vahid Kazemi, Ali Elqursh. Show, Ask, Attend, and Answer: A Strong Baseline for Visual Question Answering [EB/OL]. https://arxiv.org/abs/1704.03162, 2017-04-11/2021-11-20.

[34] Yang Zichao, He Xiaodong, Gao Jianfeng, Deng Li, Smola Alex. Stacked Attention Networks for Image Question Answering [A]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [C], New York, IEEE, 2016: 21-29.

[35] Goodfellow Ian, Bengio Yoshua, Courvile Aaron. DEEP LEARNING [M]. 赵申剑，黎彧君，符天凡，李凯译，北京：人民邮电出版社，2017.

[36] Sundararajan Mukund, Taly Ankur, Yan Qiqi. Axiomatic Attribution for Deep Networks [EB/OL]. https://arxiv.org/abs/1703.01365, 2017-03-04/2021-11-20.

附录（另起新页。若非必须，一般不需要此部分）

附录1 **CICIDS-2018特征表**

附录1 **CICIDS-2018人工特征集**

|  |  |
| --- | --- |
| 特征名 | 特征名释义 |
| fl\_dur | 流持续时间 |
| tot\_fw\_pk | 正向流的数据包总个数 |
| tot\_bw\_pk | 反向流的数据包总个数 |
| tot\_l\_fw\_pkt | 正向流中所有数据包的字节之和 |
| fw\_pkt\_l\_max | 正向流数据包中最大字节数 |
| fw\_pkt\_l\_min | 正向流数据包中最小字节数 |
| fw\_pkt\_l\_avg | 正向流数据包的平均字节数 |
| fw\_pkt\_l\_std | 正向数据包的标准偏差大小 |
| Bw\_pkt\_l\_max | 反向流数据包中最大字节数 |
| Bw\_pkt\_l\_min | 反向流数据包中最小字节数 |
| Bw\_pkt\_l\_avg | 反向流数据包的平均字节数 |
| Bw\_pkt\_l\_std | 反向数据包的标准偏差大小 |
| fl\_byt\_s | 流字节率，即每秒传输的字节量 |
| fl\_pkt\_s | 流数据包率，即每秒传输的数据包数 |
| fl\_iat\_avg | 两个流之间的平均时间 |
| fl\_iat\_std | 两个流之间的标准差时间 |
| fl\_iat\_max | 两个流之间的最大时间 |
| fl\_iat\_min | 两个流之间的最大时间 |
| fw\_iat\_tot | 正向发送的两个数据包之间的总时间，间隔时间之和 |
| fw\_iat\_avg | 正向发送的两个数据包之间的平均时间，平均间隔时间 |
| fw\_iat\_std | 正向发送的两个数据包之间的标准差时间 |
| fw\_iat\_max | 正向发送的两个数据包之间的最大时间，最大间隔时间 |
| fw\_iat\_min | 正向发送的两个数据包之间的最小时间，最小间隔时间 |
| bw\_iat\_tot | 反向发送的两个数据包之间的总时间，间隔时间之和 |
| bw\_iat\_avg | 反向发送的两个数据包之间的平均时间 |
| bw\_iat\_std | 反向发送的两个数据包之间的标准差时间 |
| bw\_iat\_max | 反向发送的两个数据包之间的最大时间 |
| bw\_iat\_min | 反向发送的两个数据包之间的最小时间 |
| fw\_psh\_flag | 在正向传输的数据包中设置 PSH 标志的次数（0 表示 UDP） |
| bw\_psh\_flag | 在反向传输的数据包中设置 PSH 标志的次数（0 表示 UDP） |
| fw\_urg\_flag | 在正向传输的数据包中设置 URG 标志的次数（0 表示 UDP） |
| bw\_urg\_flag | 在反向传输的数据包中设置 PSH 标志的次数（0 表示 UDP） |
| fw\_hdr\_len | 正向流中数据包头部的总字节数 |
| bw\_hdr\_len | 反向流中数据包头部的总字节数 |
| fw\_pkt\_s | 每秒转发数据包数 |
| bw\_pkt\_s | Number of backward packets per second |
| pkt\_len\_min | Minimum length of a flow |
| pkt\_len\_max | Maximum length of a flow |
| pkt\_len\_avg | Mean length of a flow |
| pkt\_len\_std | Standard deviation length of a flow |
| pkt\_len\_va | Minimum inter-arrival time of packet |
| fin\_cnt | Number of packets with FIN |
| syn\_cnt | Number of packets with SYN |
| rst\_cnt | Number of packets with RST |
| pst\_cnt | Number of packets with PUSH |
| ack\_cnt | Number of packets with ACK |
| urg\_cnt | Number of packets with URG |
| cwe\_cnt | Number of packets with CWE |
| ece\_cnt | Number of packets with ECE |
| down\_up\_ratio | Download and upload ratio |
| pkt\_size\_avg | Average size of packet |
| fw\_seg\_avg | Average size observed in the forward direction |
| bw\_seg\_avg | Average size observed in the backward direction |
| fw\_byt\_blk\_avg | Average number of bytes bulk rate in the forward direction |
| fw\_pkt\_blk\_avg | Average number of packets bulk rate in the forward direction |
| fw\_blk\_rate\_avg | Average number of bulk rate in the forward direction |
| bw\_byt\_blk\_avg | Average number of bytes bulk rate in the backward direction |
| bw\_pkt\_blk\_avg | Average number of packets bulk rate in the backward direction |
| bw\_blk\_rate\_avg | Average number of bulk rate in the backward direction |
| subfl\_fw\_pk | The average number of packets in a sub flow in the forward direction |
| subfl\_fw\_byt | The average number of bytes in a sub flow in the forward direction |
| subfl\_bw\_pkt | The average number of packets in a sub flow in the backward direction |
| subfl\_bw\_byt | The average number of bytes in a sub flow in the backward direction |
| fw\_win\_byt | Number of bytes sent in initial window in the forward direction |
| bw\_win\_byt | of bytes sent in initial window in the backward direction |
| Fw\_act\_pkt | of packets with at least 1 byte of TCP data payload in the forward direction |
| fw\_seg\_min | Minimum segment size observed in the forward direction |
| atv\_avg | Mean time a flow was active before becoming idle |
| atv\_std | Standard deviation time a flow was active before becoming idle |
| atv\_max | Maximum time a flow was active before becoming idle |
| atv\_min | Minimum time a flow was active before becoming idle |
| idl\_avg | Mean time a flow was idle before becoming active |
| idl\_std | Standard deviation time a flow was idle before becoming active |
| idl\_max | Maximum time a flow was idle before becoming active |
| idl\_min | Minimum time a flow was idle before becoming active |

附录2 XXXXXX

**会话样本集标签判断依据表**

作者在读期间科研成果简介

[1]\*\*\*.端到端的基于深度学习的网络入侵检测方法[J].通信技术，2022.（已录用）

致 谢

光阴似箭，日月如梭，时光荏苒，过去的一幕幕如剪影版流淌。回顾读书的这一端岁月，恍若昨日，这期间的迷茫，失望，高兴和痛苦都是值得回味，吸取经验教训的美好记忆。

本文的完成，绝非我一人的努力工作，也包含了老师和同学的帮助关心。在此，我特别要感谢我的研究生导师老师，在论文写作期间，老师给予了我们写作上的细致指导，从论文的开题到创新点的设想，为我们把关。从时间管理上给予了合理而准确的督促，因此，论文才得以顺利完成。在论文写作期间，也非常同学对我的帮助和支持，感谢同学在格式上给予我的帮助，感谢同学前期的关于卷积网络的论文分享引起我对深度学习的兴趣，也感谢师兄师姐，大家一起进行论文分享学习，互相督促，一起改进论文。另外，也非常感谢同学在项目期间对我的代码学习给予的细致指导，那时的学习真的对我帮助很大。

师姐负责组织我们新入学的同学每周开论文分享研究会，这期间的学习奠定了我之后研究的兴趣和基础，在此，也非常谢谢师姐的指导帮助。在生活中，我要特别感谢我们的辅导员老师。老师对于我们在课业之外的指导无微不至，且非常的幸苦，我们同学都十分的佩服，对老师的感谢之情无以言表。也非常谢谢老师对我们的研究生生活中关于教学方面的指导和努力工作。

此外，四川大学网络空间安全学院开设了许多理论基础课程和新兴前沿科技的课，我从中学习了基础理论知识，了解了当前的网络安全研究前沿方向，给予了我非常大的帮助，在此，我对所有上课的老师和学院献上真诚的感谢之情。

山青鸟鸣引歌行，峰绝云出落晚霞。

声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得四川大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

本学位论文成果是本人在四川大学读书期间在导师指导下取得的，论文成果归四川大学所有，特此声明。

论文作者：

指导教师：

2022年 3 月 2 日

学位论文版权使用授权书

本学位论文作者完全了解四川大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权四川大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩影或扫描等复印手段汇编学位论文。

（保密的学位论文在解密后适用本授权书）

学位论文作者签名： 导师签名：

签字日期： 年 月 日 签字日期： 年 月 日

学位论文作者毕业后去向：

工作单位： 电话：

通讯地址： 邮编：